Towards an interdisciplinary, socio-technical analysis of software ecosystem health

Tom Mens
Software Engineering Lab
University of Mons
Mons, Belgium
tom.mens@umons.ac.be

Bram Adams
MCIS Lab
Polytechnique Montreal
Montréal, Canada
bram.adams@polymtl.ca

Josianne Marsan
Faculty of Business Administration
Laval University
Québec, Canada
josianne.marsan@sio.ulaval.ca

***Abstract*—This extended abstract presents the research goals and preliminary research results of the interdisciplinary research project *SECOHealth*, an ongoing collaboration between research teams of Polytechnique Montreal (Canada), the University of Mons (Belgium) and Laval University (Canada). *SECOHealth* aims to contribute to research and practice in software engineering by delivering a validated interdisciplinary scientific methodology and a catalog of guidelines and recommendation tools for improving software ecosystem health.**

***Index Terms*—software, ecosystem, evolution, health, recommendation, prediction, survival, sustainability, resilience, socio-technical**

I. Introduction

The two-year interuniversity *SECOHealth* project[1] started on October 1, 2017. The three authors of this extended abstract are its Principal Investigators. *SECOHealth* aims to contribute to research and practice in software engineering by delivering an interdisciplinary scientific methodology and a catalog of guidelines and recommendation tools for improving software ecosystem (SECO) health. Those will enable key ecosystem actors to better monitor and control the SECO health and equip them with corrective and preventive measures to ensure their SECO's survival and sustainability. The interdisciplinary methodology used in our project will also guide other researchers in interdisciplinary projects involving open source communities or SECOs.

SECOs are large collections of interacting and interdependent software projects that share a common technological platform and that are maintained by large online communities of contributors. They pervade every aspect of human life including entertainment, health, economy, industry, politics, education and science. Commercial SECOs such as mobile app stores or the Internet-of-Things have taken over our daily lives by storm, to the extent that the functioning of our modern digitally-enabled society would be severely impacted if SECOs degrade in stability or even cease to exist.

Yet, despite the strategic importance of ensuring the overall well-being of SECOs, their health is still ill-apprehended, as SECOs are subject to constant evolution, due to an increasing pace of events (e.g., technological or environmental changes). What makes this especially challenging, is that SECOs do not have a centralised management for overseeing the ecosystem's health and survival. Instead, maintainers of SECO components need to understand and make decisions about the socio-technical impact of important events affecting SECO health and recommend corrective actions (e.g., improving SECO quality and its attractiveness to key actors). Unfortunately, there is only little support or best practices to enable SECO maintainers to perform these tasks.

[1] www.secohealth.org

II. About software ecosystem health

From a biological point of view, health can be defined as "the extent to which an organism's vital systems are performing normally at any given time" [1]. This definition can be transposed to SECO health [2] by considering a SECO as a living organism, whose constituent software projects are the vital technical systems that need to perform "normally" in order to have a healthy ecosystem, and whose community is healthy if all community members are performing normally.

SECO health problems can be very diverse in nature, and can have many different causes. For example, in March 2016, the npm ecosystem experienced the problem of a package getting unpublished, causing several thousands of transitively dependent packages to break. The underlying cause was a typical case of rage quitting, where the owner of the package decided to remove all of its packages[2]. Another documented example of rage quitting relates to "toxic" communication styles of open source communities, such as the one of the Linux Kernel community, causing a prominent developer to quit[3]; or the case where a central contributor to the bug handling community of Gentoo Linux unexpectedly left, causing a major disruption in the community's activity [3]. From a more technical point of view, typical examples of health problems are packages containing bugs or security vulnerabilities, causing potential problems in packages depending on it. The impact of the problem grows as the number of transitive dependencies on a problematic package grows.

[2] blog.npmjs.org/post/141577284765/kik-left-pad-and-npm
[3] https://slashdot.org/story/15/10/05/2031247/linux-kernel-dev-sarah-sharp-quits-citing-brutal-communications-style

III. Project Goals

SECOHealth aims at providing a scientific methodology and disciplined set of techniques to understand and control the health of software ecosystems. We adopt a socio-technical perspective since the technical and social layers of SECOs are strongly interwoven [4]. Our project aims to:

- define a conceptual model of SECO health;
- explore analogies from other scientific disciplines such as ecology and toxicology;
- determine indicators capable of measuring the different aspects of SECO health;
- determine events that affect the health of a SECO and its constituent projects;
- empirically validate these health indicators and events, both qualitatively and quantitatively;
- build and evaluate models to predict the impact of a given event on SECO health;
- build and evaluate a socio-technical dependency model to understand how health problems propagate throughout a SECO;
- propose a catalog of guidelines and recommendations for supporting SECO health.

Joining our complementary strengths in theory-driven and data-driven investigation, we will follow a mixed-methods approach [5], combining bottom-up data mining and top-down interview/survey-based research, as well as combining state-of-the-art quantitative and qualitative analysis techniques emanating from different scientific disciplines.

Under the approval of Research Ethics Committees from the participating universities, we conducted face-to-face interviews at the European Open Source Summit of the Linux Foundation (Prague, October 2017) with 17 SECO practitioners. The interviews followed the guidelines of Patton [6], with the goal of understanding what SECO health means for practitioners, what indicators they use themselves or could be used given the right data, and which events have impacted SECO health in the past.

We will operationalise the SECO health indicators into concrete metrics, and perform SECO data mining to measure and evaluate the identified health indicators. We will build and empirically validate prediction models of how SECOs will react to events, by relying on historical data from version control systems, code review and bug repositories, mailing lists and developer fora.

Based on the recent research on SECO and community health [2], [7]–[9], we will consider three high-level characteristics of health: *technical* (i.e., concerning technical software artefacts), *social* (i.e., concerning contributor communities and the relations between their members) and *phenomenological* (i.e., concerning external/internal events and their manifestation). Technical health characteristics include traditional software quality metrics, software dependency structure, software growth rate, size and frequency of software updates, bug fixes, security vulnerabilities, obsolete or deprecated components, and so on. Social characteristics include responsiveness of contributors (e.g., mean time to respond to a question, mean time to fix a bug), social network structure and its evolution (e.g. turnover rate), contributor activity and productivity, and the quality of interaction between all human stakeholders. Phenomenological health characteristics include the amount of company involvement (i.e., paid contributors), market share, presence of competing products, and so on.

With respect to the social health problem of developer turnover, we conducted an empirical study on the npm and RubyGems ecosystems. Using the statistical technique of survival analysis we identified which social or technical factors in a SECO coincide with a higher or lower probability of developer abandonment [10].

Concerning technical health, we carried out a quantitative empirical analysis of the evolution of package dependency networks for seven package distributions of varying size and age [11]. We proposed metrics to capture the growth, changeability, reusability and fragility of these dependency networks. We observed that the dependency networks tend to grow over time, while a minority of packages are responsible for most of the package updates. The majority of packages depend on other packages, but only a small proportion of packages accounts for most of the reverse dependencies. We observed a high proportion of “fragile” packages due to a high and increasing number of transitive dependencies.

IV. Interdisciplinary Research

SECOHealth will view SECOs as ecological ecosystems comprised of a population of living organisms (interdependent software projects and their interacting communities of contributors), and will produce health indicators and prediction models by drawing inspiration from well-known principles and theories from other disciplines, such as the notion of biodiversity in ecology [12], or the notion of toxicity in toxicology [13], [14].

SECO health needs to be studied at different levels of granularity since the health of the SECO as a whole depends on the health of its social and technical components, and vice versa. At a *micro-level* of analysis (i.e., within and between individual projects of a SECO), we will explore the impact of *toxicity*, arguing that certain behaviour and interactions in the SECO community can be toxic to not only the individual software projects, but even to the SECO as a whole, and hence can jeopardize its health and sustainability. Examples of possible toxic social behaviour may consist of deviant or aggressive behaviours, for example in the form of flame wars as a reaction to bad quality code contributions [13]. One promising way to assess such toxicity is by measuring *social debt* [15], i.e., social interactions between SECO members that have been strained due to time pressure or lack of attention, and at some point might blow up and cause friction within the community of developers involved in a software project.

At a *macro-level*, we will study how health problems of SECO components evolve and propagate to others. Among others, we will test the principle of *biodiversity* by analysing to which extent the SECO’s *resilience* decreases when its diversity decreases. By resilience we refer to the ecosystem's capacity of resisting to disturbances, or recovering from a perturbation quickly. Diversity will be analysed according to a variety of factors (e.g., geographical, activity-related, time-related, gender-related, artefact-related). Some of these factors have been shown to have positive effects on software health. For example, gender diversity has been shown to have a positive effect on the productivity of GitHub teams [16].

V. Related Work and Related Projects

SECOHealth can be considered as a successor project of the *ECOS* project (Ecological Studies of Open Source Software Ecosystems) [17] that was carried out from 2012 till 2017. As part of that project, we carried out multiple empirical studies of the evolution of open source software ecosystems (e.g., Gnome [18], CRAN [19]–[21], Debian [22], npm and RubyGems [23]–[25]).

The *SECOHealth* members are actively involved in the CHAOSS (Community Health Analytics of Open Source Software) initiative of the Linux Foundation. Metrics Committee. Its goal is to define, implement and assess metrics for open source community health and sustainability. While CHAOSS' initial focus is on metrics at the level of individual software projects, *SECOHealth* will focus on ecosystem-wide health metrics. To operationalise our metrics, we aim to use Bitergia's open source GrimoireLab tool chain for software development analytics[4], which is one of the tool suites actively supported by CHAOSS.

Acknowledgment

This research is carried out in the context of FRQ-FNRS collaborative research project R.60.04.18.F "SECOHealth" and FNRS Research Credit J.0023.16 "Analysis of Software Project Survival". We thank our external project collaborators Kevin Carillo (Toulouse University, France), Damian Tamburri (Politecnico di Milano, Italy), Gregorio Robles (Universidad Rey Juan Carlos, Spain) and Bogdan Negoita (HEC Montréal, Canada) for actively participating to this project.

References

[1] X. Wang and S. Lantzy, "A systematic examination of member turnover and online community health," in *Int'l Conf. Information Systems (ICIS)*, 2011.

[2] S. Jansen, "Measuring the health of open source software ecosystems: Beyond the scope of project health," *Information and Software Technology*, vol. 56, no. 11, pp. 1508 – 1519, 2014, special issue on Software Ecosystems.

[3] M. S. Zanetti, I. Scholtes, C. J. Tessone, and F. Schweitzer, "The rise and fall of a central contributor: Dynamics of social organization and performance in the Gentoo community," in *International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, 2013, pp. 49–56.

[4] T. Mens, "An ecosystemic and socio-technical view on software maintenance and evolution," in *Int'l Conf. Software Maintenance and Evolution (ICSME)*, 2016, pp. 1–8.

[5] R. B. Johnson and A. J. Onwuegbuzie, "Mixed methods research: A research paradigm whose time has come," *Educational Researcher*, vol. 33, no. 7, pp. 14–26, 2004.

[6] M. Q. Patton, *Qualitative research and evaluation methods : integrating theory and practice*. SAGE Publications, 2015.

[7] F. Fotrousi, S. A. Fricker, M. Fiedler, and F. Le-Gall, "KPIs for software ecosystems: A systematic mapping study," in *Int'l Conf. Software Business (ICSOB)*. Springer, 2014, pp. 194–211.

[8] J. Y. Monteith, J. D. McGregor, and J. E. Ingram, "Proposed metrics on ecosystem health," in *Int'l Workshop on Software-defined Ecosystems (BigSystem)*. ACM, 2014, pp. 33–36.

[9] K. Manikas and D. Kontogiorgos, "Characterizing software activity: The influence of software to ecosystem health," in *Int'l Workshop on Software Ecosystems (IWSECO), European Conf. Software Architecture Workshops*, 2015.

[10] E. Constantinou and T. Mens, "An empirical comparison of developer retention in the RubyGems and npm software ecosystems," *Innovations in Systems and Software Engineering*, vol. 13, no. 2-3, pp. 101–115, 2017.

[11] A. Decan, T. Mens, and P. Grosjean, "An empirical comparison of dependency network evolution in seven software packaging ecosystems," *Empirical Software Engineering*, 2018.

[12] T. Mens, M. Claes, P. Grosjean, and A. Serebrenik, *Studying Evolving Software Ecosystems based on Ecological Models*. Springer, 2014, pp. 297–326.

[13] K. Carillo and J. Marsan, ""The dose makes the poison" - exploring the toxicity phenomenon in online communities," in *Int'l Conf. Information Systems (ICIS)*, 2016.

[14] K. Carillo, J. Marsan, and B. Negoita, "Exploring the biosphere – towards a conceptualization of peer production communities as living organisms," in *SIGOPEN Pre-ICIS 2017 Workshop on Open Phenomena, Association for Information Systems (AIS)*, Seoul, South Korea, 2017.

[15] D. A. Tamburri, P. Kruchten, P. Lago, and H. v. Vliet, "Social debt in software engineering: insights from industry," *Journal of Internet Services and Applications*, vol. 6, no. 1, May 2015.

[16] B. Vasilescu, D. Posnett, B. Ray, M. G. J. van den Brand, A. Serebrenik, P. T. Devanbu, and V. Filkov, "Gender and tenure diversity in GitHub teams," in *Int'l Conf. Human Factors in Computing Systems (CHI)*. ACM, 2015, pp. 3789–3798.

[17] T. Mens, M. Claes, and P. Grosjean, "ECOS: Ecological studies of open source software ecosystems," in *IEEE Int'l Conf. Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 403–406.

[18] B. Vasilescu, A. Serebrenik, M. Goeminne, and T. Mens, "On the variation and specialisation of workload: A case study of the Gnome ecosystem community," *Empirical Software Engineering* vol. 19, pp. 955–1008, 2014.

[19] D. M. Germán, B. Adams, and A. E. Hassan, "The evolution of the R software ecosystem," 2013, pp. 243–252.

[20] M. Claes, T. Mens, and P. Grosjean, "On the maintainability of CRAN packages," in *IEEE Int'l Conf. Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp. 308–312.

[21] A. Decan, T. Mens, M. Claes, and P. Grosjean, "When GitHub meets CRAN: An analysis of inter-repository package dependency problems," in *Int'l Conf. Software Analysis, Evolution, and Reengineering (SANER)*, 2016.

[22] M. Claes, T. Mens, R. D. Cosmo, and J. Vouillon, "A historical analysis of Debian package incompatibilities," in *Working Conf. Mining Software Repositories (MSR)*, 2015, pp. 212–223.

[23] E. Constantinou and T. Mens, "Social and technical evolution of software ecosystems: A case study of Rails," in *Int'l Workshop on Software Ecosystems (IWSECO), European Conf. Software Architecture Workshops*, 2016.

[24] A. Decan, T. Mens, and M. Claes, "An empirical comparison of dependency issues in OSS packaging ecosystems," in *Int'l Conf. Software Analysis, Evolution and Reengineering (SANER)*, 2017, pp. 2–12.

[25] E. Constantinou and T. Mens, "Socio-technical evolution of the Ruby ecosystem in GitHub," in *Int'l Conf. Software Analysis, Evolution and Reengineering (SANER)*, 2017, pp. 34–44.

[4]http://grimoirelab.github.io