HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jides

The importance of socio-technical resources for software ecosystems management

Thaiana Lima[*], *Rodrigo Pereira dos Santos*, *Jonice Oliveira*, *Cláudia Werner*

System Engineering and Computer Science Department, PESC/COPPE/UFRJ – Federal University of Rio de Janeiro, Zip Code 21945-970 – Rio de Janeiro, Brazil

HIGHLIGHTS

- A survey on the relevance of socio-technical resources for SECOs based on literature.
- Expert opinion on the main roles, activities and artifacts in ecosystem platforms.
- Discussion of socio-technical resources in two real ecosystems, BPS Portal and GitHub.
- A second survey on the usefulness and ease of use of two real ecosystem platforms.
- The top 12 socio-technical resources evaluated in two real SECO platforms.

ARTICLE INFO

Article history:
Received 22 March 2016
Received in revised form
25 October 2016
Accepted 31 October 2016
Published online 16 November 2016

Keywords:
Software ecosystems
Socio-technical resources
Component repositories
Software reuse

ABSTRACT

Software Ecosystem (SECO) is often understood as a set of actors interacting among themselves and manipulating artifacts with the support of a common technology platform. Usually, SECO approaches can be designed as an environment whose component repository is gathering stakeholders as well as software products and components. By manipulating software artifacts, a technical network emerges from interactions made over the component repository in order to reuse artifacts, improving code quality, downloading, selling, buying etc. Although technical repositories are essential to store SECO's artifacts, the interaction among actors in an emerging social network is a key factor to strengthen the SECO's through increasing actor's participation, e.g., developing new software, reporting bugs, and communicating with suppliers. In the SECO context, both the internal and external actors keep the platform's components updated and documented, and even support requirements and suggestions for new releases and bug fixes. However, those repositories often lack resources to support actors' relationships and consequently to improve the reuse processes by stimulating actors' interactions, information exchange and better understanding on how artifacts are manipulated by actors. In this paper, we focused on investigating SECO as component repositories that include socio-technical resources. As such, we present a survey that allowed us to identify the relevance of each resource for a SECO based on component repositories, initially focused on the Brazilian scenario. This paper also describes the analysis of the data collected in that survey. Information of other

Peer review under responsibility of Qassim University.

[*] *Corresponding author.*

E-mail addresses: thaiana@cos.ufrj.br (T. Lima), rps@cos.ufrj.br (R.P.d. Santos), jonice@dcc.ufrj.br (J. Oliveira), werner@cos.ufrj.br (C. Werner).

http://dx.doi.org/10.1016/j.jides.2016.10.006

SECO elements extracted from the data is also presented, e.g., the participants' profile and how they behave within a SECO. As an evolution of our research, a study for evaluating the availability and the use of such resources on top of two platforms was also conducted with experts in collaborative development in order to analyze the usage of the most relevant resources in real SECO's platforms. We concluded that socio-technical resources have aided collaboration in software development for SECO, coordination of teams based on more knowledge of actor's tasks and interactions, and monitoring of quality of SECOs' platforms through the orchestration of the contributions developed by external actors.

1. Introduction

Implementing software reuse can increase code quality, productivity and time-to-market since a component is not built from scratch—the same applies to other artifacts related to the software development, e.g., templates, documents and architecture. The traditional strategy of building isolated, monolithic systems within the organization is fading away in face of the component-based systems [1]. Those systems implement software reuse by developing software components to be integrated into the systems. The variability in a product depends on its potential components and can generate a product line. Product lines develop different versions of the same product according to the possible variability [2]. As an evolution of a product line, a Software Ecosystem (SECO) represents the extrapolation of organizational limits [3], facing a much larger set of elements such as social and business issues, orchestration of external actors, and management and monitoring of multiple software products and services. They implement many product lines at the same time focusing at technical, social and business dimensions of software development [4].

In order to support Software Reuse, a well-known and applied technique to help developers finding components consists of implementing a repository of reusable software artifacts. This type of repository stores components and related information, e.g., documents, architecture, source code etc. [1]. On top of such repositories, SECOs arise from the interactions among actors, i.e., developers and users (either internal or external). A SECO is created from a common technological platform focused on software products and services [2], contributing to explore the interorganizational reuse [2]. As a dynamic environment, it is important to boost actors' participation and artifacts' publishing, as well as community's discussion to maintain the SECO platform alive. As such, since an organization stops building software products isolated from other companies and starts seeking partnerships, opening business strategies go beyond organizational borders and encounter an ecosystem made up of various organizations [2]. Thus, it is relevant to study a SECO as a set of platforms, actors and artifacts/information within a software supply network [3,4]. In doing so, it is possible to analyze the evolution of SECO's software identifying potential investments in new releases or fixes, identify demands from the community. In addition, the role of external developer changes the traditional development management strategy. The keystone (i.e., organization that is responsible for the SECO platform) does not have complete control over an external developer. They can leave the SECO at any time (taking information with them) or enter (requesting information). This, it becomes a necessity to monitor the SECO in order to better understand its behavior and evolution. The keystone organization is mainly responsible for monitoring the SECO, evaluating it, making decisions, and taking actions [5].

In this scenario, the interactions among actors lack effective attention to encourage social relationships [6]. Due to different types of relationship among actors and artifacts like 'communicate with' and 'depends on', the existing networks are neither solely social nor technical; they include both actors and artifacts. Exploring socio-technical relationships can reveal information from the SECO that was too spread out to be organized, e.g., community's tendency and demands can be extracted after analyzing frequency of terms from the communications among actors. Those information contribute to the software development from the point of view of the keystone that can prioritize functions and bug fixes according to the community data; the developer that now have information of dependency relationships, e.g., helping to select a component; the user that can better understand the product based on the community's relationship information, e.g., information of use, reported problems, and technology dependency; and other benefits of comprehending how such elements are interacting and influencing each other. In turning the relationships in a SECO more explicit, its central platform and keystone can analyze the SECO as a set of integrated elements using the network drawn by the elements and its relationships, revealing new structural and influence information.

Aiming to support the social networks created from a SECO, it is important to provide social resources that foster actors' interaction and also include resources that allow software artifacts manipulation, i.e., the socio-technical resources. With the purpose of identifying the most relevant and suitable socio-technical resources for the SECO platform management, a survey was conducted with Brazilian experts in SECO, collaborative systems and distributed software development. This study allowed us to organize a set of social and technical resources presented in the literature as well as to analyze them in the context of an existing Brazilian government open source SECO—the Brazilian Public Software (BPS) Portal [7]. According to a broad systematic review on