

Chapter 56

A Systematic Mapping Study on Requirements Engineering in Software Ecosystems

Aparna Vegendla

Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Anh Nguyen Duc

Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Shang Gao

Örebro University, Örebro, Sweden

Guttorm Sindre

Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ABSTRACT

Software ecosystems (SECOs) and open innovation processes have been claimed as a way forward for the software industry. A proper understanding of requirements is as important for SECOs as for more traditional ones. This article presents a mapping study on the issues of RE and quality aspects in SECOs. Our findings indicate that among the various phases or subtasks of RE, most of the SECO specific research has been accomplished on elicitation, analysis, and modeling. On the other hand, requirement selection, prioritization, verification, and traceability has attracted few published studies. Among the various quality attributes, most of the SECOs research has been performed on security, performance and testability. On the other hand, reliability, safety, maintainability, transparency, usability attracted few published studies. The article provides a review of the academic literature about SECO-related RE activities, modeling approaches, and quality attributes, positions the source publications in a taxonomy of issues and identifies gaps where there has been little research.

DOI: 10.4018/978-1-7998-3016-0.ch056

INTRODUCTION

The rapid pace of technological changes and the competitive race for quick product release are driving many companies to look for new ways to deliver software. Software product lines (SPLs) are one step towards making software development more efficient (Bosch & Bosch-Sijtsema, 2010). In SPL, a set of business units in an organization could develop the products through collaboration by sharing a common technological platform, and by reusing much of the software between different versions and variants of the product. Over the past decade, companies have been transitioning their SPLs to software ecosystems (SECOs) to open their platforms for external software providers (Bosch, 2009). The goal is to rapidly develop new capabilities and foster innovations unforeseeable by the platform's original designers (Jansen & Cusumano, 2013). The SECOs are multi-disciplinary systems inspired from business and natural ecosystems. Manikas and Hansen define software ecosystem as "...the interaction of a set of actors on top of a common technological platform that results in a number of software solutions or services..." (Manikas & Hansen, 2013) (p. 1297). For example, Google controls the Android platform while external developers can build applications that are distributed to Android users via the Google Play store. Thus, Google has collaborated with external developers to build functionality in the form of available applications. In contrast to the software development in an individual organization, SECO includes the software development by several organizations through collaboration and competition (Bosch-Sijtsema & Bosch, 2015). For instance, Microsoft made the PowerShell tool built on Microsoft .NET as an open source product to keep the developers interested in the Windows platform while Google released Cloud Tools for PowerShell to make Google's cloud more attractive to .NET developers. Either way, both Google and Microsoft co-create value through collaboration and competition.

Despite the perceived advantages of SECOs, transitioning to SECOs may have challenges with communication barriers between parties due to the dispersion of SECO members. On the other hand, providing the open platform to external actors raises the conflicts of interest when negotiating requirements. One of the main issues is inconsistency and variability in stakeholders' requirements. Requirements engineering (RE) is essential for SECO's to involve stakeholders early in the development to understand requirements and use cases. The impact of changes can be analyzed and documented through a model of the system (Hull, Jackson, & Dick, 2011) during the early stage of RE. Modeling can aid the stakeholders of a SECO to make sustainable relations among the actors when they negotiate their common interests (i.e. requirements) for the software. The obvious question to ask is whether the RE process used for traditional systems can cope with the context of SECO's? How can the RE process best be conducted when developing multi-organizational, socio-technical systems like SECOs? Can adaptation of existing traditional approaches used in designing of technical system aids modeling of SECOs or does SECO require a new approach? Our research questions are: RQ1: What RE activities have been studied specifically in relation to software ecosystems earlier? RE activities address, how are the requirements elicited, analyzed, documented, validated, and traced. RQ2: How are non-functional requirements considered in the context of SECOs? A number of challenges in SECO's have been discussed in the literature (Serebrenik & Mens, 2015), we focus on challenges specific to non-functional requirements in SECOs.

The remainder of the paper is structured as follows. Section 2 provides the background of SECOs and RE. In Section 3, we describe the research method used for conducting a literature review for the paper. Section 4 analyzes the results. Section 5 provides discussion of research gaps identified in the existing literature on RE in SECOs follows with conclusion in section 6.

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/a-systematic-mapping-study-on-requirements-engineering-in-software-ecosystems/261076

Related Content

Girls and Computers - Yes We Can!: A Case Study on Improving Female Computer Confidence and Decreasing Gender Inequity in Computer Science with an Informal, Female Learning Community

Misook Heo and L. Monique Spradley-Myrick (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 1126-1143).

www.irma-international.org/chapter/girls-computers-yes-can/62502

Effort Estimation Model for each Phase of Software Development Life Cycle

Sarah Afzal Safavi and Maqbool Uddin Shaikh (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 238-246).

www.irma-international.org/chapter/effort-estimation-model-each-phase/62445

Role of Free and Open Source GIS in River Rejuvenation

Smart Kundassery and Babu C. A. (2018). *Emerging Trends in Open Source Geographic Information Systems* (pp. 195-217).

www.irma-international.org/chapter/role-of-free-and-open-source-gis-in-river-rejuvenation/205161

Cyber-Security Concerns With Cloud Computing: Business Value Creation and Performance Perspectives

Ezer Osei Yeboah-Boateng (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 995-1026).

www.irma-international.org/chapter/cyber-security-concerns-with-cloud-computing/203545

Leveraging Business Model Innovation in the International Space Industry

Alessandra Vecchi and Louis Brennan (2020). *Disruptive Technology: Concepts, Methodologies, Tools, and Applications* (pp. 625-643).

www.irma-international.org/chapter/leveraging-business-model-innovation-in-the-international-space-industry/231209