



Exploiting Repositories in Mobile Software Ecosystems from a Governance Perspective

Awdren de Lima Fontão¹ · Rodrigo Pereira dos Santos² · Arilo Claudio Dias-Neto¹

Published online: 27 May 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Software organizations that maintain mobile platforms (e.g. Apple, Google, and Microsoft) open their infrastructure to attract and engage external developers to meet the demands of users. This scenario has been investigated as Mobile Software Ecosystem (MSECO). In MSECO, an organization provides content reuse repositories where users can search, download and evaluate apps, or where developers can get support materials. Such repositories are commonly inspired by reuse mechanisms and help an organization to control and evolve the whole ecosystem to keep it vibrant. In this scenario, it is important to understand how to govern MSECO as well as how reuse mechanisms can support this goal. As such, we investigated governance strategies that emerge from content reuse repositories based on four steps: (1) designing a governance process; (2) performing an initial evaluation of this process with managers who are experts in the main existing MSECO (Apple, Google, Nokia, Microsoft, and Samsung); (3) interviewing 18 MSECO managers to collect opinion about the process; and (4) conducting an exploratory analysis of the MSECO reuse repositories' structure based on the proposed governance process. Three types of content reuse repositories (Apps' Store, Developer's Central, and Apps' Management) were analyzed. As a result, this research allowed us to define a governance process from a repository perspective containing 11 activities.

Keywords Software ecosystems · Mobile application · Process · Model · Governance · Reuse · Repositories

1 Introduction

The new software development model based on mobile platform has affected the way organizations deliver their products and services and how they manage resources for market expansion. In other words, the increasing advance of mobile technology based on “apps” has fostered the mobile device market, where large companies such as Apple, Google and Microsoft promote actions for leveraging applications development. In this scenario, external developers are a key

resource for organizations that want to meet user demands (Santos and Werner 2012). It comprises the following dimensions: *technical* (software development process), *social* (interaction between organization and developers), and *business* (balance between organization's goals and developers' expectations). In the mobile platform context, it is known as a Mobile Software Ecosystem (MSECO). MSECO is a cooperative evolutionary system – similar to biological ecosystem – that comprises several elements, for example: common technological platform (or just platform), mobile applications, mobile application store, evangelists, users, keystone or central organization, and external developers (Fontão et al. 2015). MSECO context allows an organization to play in the mobile industry with use of API, code reuse, market analysis, and tools (Fontão et al. 2015b).

In a specific MSECO, external developers exchange ideas and support community portals through some social mechanisms. They also collaborate with a keystone (who maintains a technological platform) since they produce mobile applications or any other artifact (e.g. code snippets and wiki articles) to support mobile application development and evolution (Fagerholm et al. 2015). Developers also can reuse artifacts available within the MSECO. For example, a developer who

✉ Awdren de Lima Fontão
awdren@icomp.ufam.edu.br

Rodrigo Pereira dos Santos
rps@uniriotec.br

Arilo Claudio Dias-Neto
arilo@icomp.ufam.edu.br

¹ Institute of Computing, Federal University of Amazonas, Manaus, AM, Brazil

² Department of Applied Informatics, Federal University of the State of Rio de Janeiro, Rio de Janeiro, RJ, Brazil

needs to build a similar feature in his/her mobile application can reuse a code template. As such, a sample project can be the starting point for a new mobile application, e.g. projects available at the official website of a specific MSECO such as *Android Developers*.¹ Based on this mindset, some authors argue that MSECO is an evolution of software reuse (Bosch 2009) (McGregor 2010)(Santos and Werner 2011). Software reuse, i.e. “*the use of existing engineering knowledge and artifacts to build new software systems*” (Frakes and Fox 1996), is considered a key element to delivery high quality software in time and on budget over the entire software development, maintenance, and evolution lifecycle. In MSECO, software reuse explores the social dimension: developer communities produce mobile applications from existing artifacts and then reuse and mature existing knowledge based on interactions and content sharing via MSECO portals.

Since artifacts can be reused in MSECO, they should meet some platform quality criteria to keep the ecosystem vibrant. To do so, content reuse repositories can support apps’ search, download and evaluation, and help developers get MSECO materials. Such repositories attract developers/users and contribute to the ecosystem resilience and productivity (Martin et al. 2017). For this reason, MSECO organizations provide governance strategies driven by content reuse repositories to support mobile application development and management, including learning resources and project hosting (Genc-Nayebi and Abran 2017). Examples of governance strategies involve guidelines, values, practices, and standards. In this scenario, interactions between MSECO elements and repositories increase complexity in coordinating and maintaining a specific MSECO through governance strategies towards a vibrant ecosystem. When an organization in an ecosystem is defining governance strategies, management commitment is a key enabler for any reuse program (Santos et al. 2017). This commitment supports reuse processes (Stol and Fitzgerald 2015) and affects organizational change. As such, in-depth studies on approaches to support external developer activities in mobile application design, development and analysis are required. Such studies should address artifacts management based on content reuse and architecture opening mechanisms for those participants (Fontao et al. 2015b).

In this work, we investigated MSECO governance strategies that emerge from content reuse repositories. To do so, we designed and analyzed a governance process and evaluated it with managers who work(ed) in the main MSECO organizations (i.e. Apple, Google, Microsoft, Nokia, and Samsung) to identify the main governance activities, artifacts, and roles. Next, we performed interviews with 18 experienced managers to refine the governance process. Then, we analyzed the main MSECO reuse repositories identified in our studies (Apps’ Store, Developer’ Central, and Apps’ Management) to

understand how reuse mechanisms and governance strategies have supported keystone and external developers to contribute to MSECO expansion.

This paper is organized as follows. Section 2 presents the background regarding MSECO and governance. Related work is discussed in Section 3. Section 4 presents the proposed MSECO governance process as well as a study with experts that allowed us to refine it. Section 5 presents a study with experienced managers to evaluate the proposed MSECO governance process. Then, Section 6 presents a study with researchers to explore content reuse repositories as an instrument to adapt governance strategies to keep the MSECO vibrant as well as to attract developers/users. Finally, Section 7 concludes the paper and points out future work.

2 Background

In this section, we discuss the main concepts regarding software ecosystems, mobile software ecosystem, and ecosystem governance.

2.1 Mobile Software Ecosystem (MSECO)

Relationships among components, infrastructures and services developed by large companies and external developers have changed software development mindset from a single product to a large ecosystem platform. In this scenario, software products’ suppliers and consumers create components such as applications, extensions and plug-ins in a collaborative way to generate value (Jansen and Cusumano 2012) (Valença and Alves 2017). As a consequence, new business models have emerged in software engineering to: redefine roles and patterns of collaboration and innovation; support complex networks of organizations or communities; and cope with ecosystem platform engineering and management (Hanssen 2012) (Manikas 2016). In this context, a software ecosystem can be defined as a collection of software products that have a certain degree of symbiotic relationship (Messerschmitt and Szyperski 2003). The main elements are described by Jansen et al. (2013):

- *Keystone*: organization that provides a technological platform, sets standards and practices, and increases the value through keeping a large number of contributors. A keystone maintains a stable and predictable set of common assets used by external developers to build their own contributions within the ecosystem; and
- *Contributor*: external organizations or developers who use a keystone’s technological platform to generate value-based software business from well-known standards in a specific ecosystem. There are some roles, such as *hedger* (who plays in several ecosystems to minimize

¹ <https://developer.android.com>

risks), *disciples* (who recently adopted the platform and published applications), and *influencers* (who change platform characteristics, organize conferences and create communities).

In order to reach a growing number of users, keystones have sought alternatives to expand their ecosystems through supporting new technologies and processes to ease application development. As such, innovation processes are adopted since integration and use of third-party contributions play an important role in software development (Neto et al., 2017). For example, to help users personalize and enjoy all mobile device features, several types of applications have been developed for different platforms (Ascate et al. 2017) (Pant and Yu 2017). In order to maximize application production, keystones have adopted technologies that ease user interaction to enhance the use of several applications (Fontão et al. 2015b). Large companies such as Google, Apple and Microsoft raise their ecosystems from domain specific repositories – Google Play, App Store and Windows Store, respectively.

A software ecosystem focused on mobile applications is known as MSECO (Mobile Software Ecosystem). MSECO consists of a cooperative evolutionary system comprising mobile applications, developers and users that form complex relationships and create market niches similarly to biological ecosystems (Lin and Ye 2009) (Fontão et al. 2015). An MSECO important element is the mobile application store, i.e. a repository (seen as the ecosystem platform) with mechanisms to publish, store, search, retrieve and catalog applications. In MSECO, organizations need to add value to their products (i.e., mobile devices and applications available in application stores). However, these organizations can hardly meet the entire demand of society only based on their internal structure (Bosch 2009) (Valença and Alves 2017).

MSECO development process mainly involves developers, their own mobile applications, and application stores (Bentley and Lim 2012a). Other elements involved in this process are: mobile application acceptance criteria, technological platform, monetization strategies, marketing, user reviews, partners, resources, documentation to support community activities, and frameworks (Jansen and Cusumano 2012). Fontão et al. (2015) present specific elements that form an MSECO:

- *Platform*: generic term for standard system architecture, communication protocol, or any fundamental, shared knowledge. A platform is a base on which several technical MSECO elements are built upon. It provides a support for mass customization;
- *Users*: represent the MSECO target. They are responsible for acquiring content from application stores and creating new forms of interaction. They download and use mobile applications, and provide feedback to developer. They also can determine their favorite mobile application

features. The number of active users in an ecosystem reflects user satisfaction;

- *Developer*: responsible for providing MSECO ideas and performing mobile application development to reach user requirements. They can be an *individual* or an *organization*;
- *Community*: a network of collaboration and activity coordination within an ecosystem, composed by internal and external actors. Some specific communities were identified, such as users' community, developers' community and experts' community – the latter helps developers to extend and integrate MSECO products based on training/mentoring services;
- *Mobile Applications (apps)*: artifacts produced by developer(s) and acquired by users. They consist of basic software units being grouped and categorized. A mobile application must meet a minimum quality standard required by an application store;
- *Mobile Applications Store (app store)*: a distribution channel used to publish, store, search and retrieve mobile applications. It is a high competitive marketplace where developers provide software that will be used by a large number of users; and
- *Evangelist*: leads trainings, lectures and development competitions to attract and support new external developers to a specific MSECO. He/She can be part of an experts' community, i.e. he/she is a domain expert who knows software development activities in a specific ecosystem.

There are at least two repositories in an MSECO: Apps' Store and Material Support Portal. Apps' Stores are democratic spaces that allow users (developers) access (publish) mobile applications over the world (Bentley and Lim 2012b). In turn, Material Support Portals feed application development, including platform tools, documentation, tutorials, videos, and sample codes, and technical support by evangelists (Fontao et al. 2015a). Developers' communities work directly on the mentioned repositories (Ahasanuzzaman et al. 2016) (Fontão et al., 2017) so that the ecosystem approach can be used to reduce development/maintenance costs (Manikas 2016). To do so, a keystone can monitor the ecosystem to keep it balanced from an interplay of all MSECO elements promoted by governance strategies (Dhungana et al. 2010) (Alves et al. 2017).

2.2 Governance in MSECO

The governance is defined as the way an organization is managed, including responsibilities and decision-making processes (Eckhardt et al. 2014). Governance involves assignment of roles and decision rights, as well as measures and policies that enable continuous assessment. Successful management and

monitoring of ecosystem remain big challenges for software ecosystem practitioners (Constantinou and Mens 2017). This is partly because the software ecosystem research and practice still lack proper management theories, tool support, and consolidated experience in this field (Manikas 2016).

Alves et al. (2017) define software ecosystem governance mechanisms as managerial tools to support players in a software ecosystem towards influencing the ecosystem health. Ecosystem health refers to the sense that a software ecosystem is functioning well. Specific measurements may be introduced to get an overview of the ecosystem status while at the same time allow ecosystem comparison. In this scenario, there are three main categories of governance mechanisms: *Value Creation* – to generate and distribute value; *Coordination* – to maintain consistency/integration of activities, relationships, and structures of the ecosystem; and *Organizational Openness and Control* – to capture the tension between open and closed models.

Governing MSECO requires a slight balance of control between platform provider and external developers (Song et al. 2018). Moreover, a well-chosen platform provides considerable competitive benefits, while a poorly-chosen one puts them at a disadvantage. In this scenario, Valença and Alves (2017) point out the need for understanding how platform governance affects MSECO innovation. In other words, understanding how the implementation of specific governance mechanisms affects the success of an ecosystem and its underlying enterprise platform is an exciting problem for researchers in the field (Alves et al. 2017).

Baars and Jansen (2012) state that ecosystem governance can help a company achieve its goals, make better use of available resources and can ultimately lead to increasing revenue and lower risks. However, since it is a relatively new field (Manikas 2016) (Mäenpää et al. 2017), many organizations do not know how to effectively manage their ecosystem, or even how to make their ecosystem ready for a governance strategy. Another point indicated by Schreieck et al. (2016) is the difficulty in evaluating how data is used to govern platform ecosystems in practice (and how to generalize the findings). Therefore, research on ecosystem governance can help scholars and practitioners to address a topic that is highly relevant in practice (Axelsson and Skoglund 2016).

Finally, proper formalization for ecosystem governance is lacking. There are several challenges to overcome in the perspective of ecosystem-driven organizations (Manikas 2016), e.g. developers' attraction and engagement. There is also a need for a common vocabulary in ecosystem governance, practical guidance, and developer governance strategies.

3 Related Work

Regarding information needs in ecosystems, Haenni et al. (2013) investigated the nature of information needs from the

perspective of software developers who work in larger ecosystems' projects. To do so, the authors interviewed several developers. Results showed that developers want to know: how their code fits to the ecosystem platform; statistics about the use of their software; how to subscribe to a mailing list to keep up-to-date with a given issue; how to access source code repositories; and how to use social media tools to visualize and publish news about their projects. Developers also claimed for available public support, documentation, real sample code, and APIs. However, the authors do not mention how should be the design and availability of the artifacts that will be consumed and extended by external developers.

Wareham et al. (2014) identified three constituent tensions of the stability-evolvability equilibrium in ecosystems that should be managed: standard-variety, control-autonomy, and collective-individual. The authors also analyzed software ecosystem governance to accommodate such tensions based on increasing and decreasing variance simultaneously within (and between) different tensions and constituents. The ecosystem governance can be designed to harness tensions as enabling forces that serve the overall platform's needs. However, the authors did not analyze the major roles involved in an ecosystem governance process and how these actors can work to keep the ecosystem healthy.

Manikas et al. (2015) argue that decisions related to governance can influence the ecosystem health and can result in fostering the success or greatly contributing to the ecosystem failure. Based on existing literature of software ecosystem governance and IT governance, the authors proposed the decomposition of software ecosystem governance into three activities: input or data collection (e.g. measures, information sources), decision making (e.g. data processing, scenario interpretation, alternative actions and their impact), and applied actions. Five decision areas (Principles, Actor Interaction, Software Interaction, Platform and Ecosystem Business, and Products) were identified as well as also four archetypes (Monarchy, Collective, Federal, and Anarchy) describing the way decisions are taken for each situation. Despite the work on understanding how governance can influence a healthy ecosystem, the authors did not explore the structure of internal repositories in the data collection activity.

Sadi et al. (2015) proposed a generic approach built upon Android and iOS ecosystems to: identify types of developers, analyze technical and non-technical requirements for collaborations; and derive alternative solutions for designing an appropriate collaboration. The authors found out that Android developers choose an open-source platform to cultivate intrinsic motivations, such as skills development and reputation enhancement. Regarding iOS developers, financial gain is the main requirement to sustain a collaborative relationship between developers and Apple. This study focuses on developers' objectives and decision criteria but does not provide specific guidance on how these activities can be performed.

To demonstrate the feasibility of the proposed guidelines, experimentation in real case studies is required.

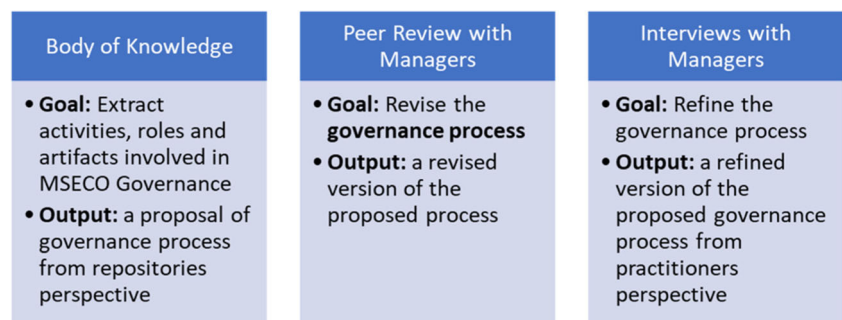
Manikas et al. (2016) focused on the creation of ecosystems and proposed a process for designing, developing, and establishing software ecosystems based on three basic steps and a set of related activities. The authors observed that ecosystems typically emerge from either a company deciding to allow development on their product platform or from a successful open source project. Their approach was demonstrated in two case studies in which ecosystems can emerge from more than one technological infrastructure (ecosystem platform). Despite the analysis, this paper does not adequately discuss software ecosystem governance, being useful to summarize different ecosystems’ characteristics and scopes.

Based on the above studies and literature reviews on software ecosystems (Barbosa et al. 2013) (Axelsson and Skoglund 2016) (Manikas 2016) (Franco-Bedoya et al. 2017), MSECO (Fontão et al. 2015) and ecosystem governance (Alves et al. 2017), there is no governance approach from the perspective of repositories, either internal to an MSECO (e.g. Apps’ Stores and Material Support Portals) and external to MSECO (e.g., GitHub and Stack Overflow). On the other hand, a discussion on which artifacts are most relevant from the organizational point of view and in which repositories the artifacts should be made available is performed in our study. Based on this critical gap, in the next section we present research steps to investigate how governance takes place in MSECO from the perspective of repositories.

4 MSECO Governance Approach

For the definition of the governance approach from the perspective of repositories, we applied the research methodology shown in Fig. 1. It started with the execution of a literature study to ground a body of knowledge for the extraction of activities related to MSECO governance and definition of our proposed approach. Next, we performed a peer review study with 2 experienced MSECO managers to revise the proposed approach. Finally, we conducted a cycle of interviews with 18 managers to evaluate the approach.

Fig. 1 Research methodology



4.1 Analyzing MSECO Governance Process

In order to support governance strategies from content reuse repositories, we proposed an initial version of an MSECO governance process based on the knowledge body and peer review with two managers as described in Fig. 1. The MSECO governance process was grounded in a systematic mapping study previously conducted and published in (Fontão et al. 2015). The objective of this process is to prepare, manage and coordinate some MSECO elements – i.e. developer, evangelist and mobile application – and its relationships (Fontão et al., 2016). It also seeks to provide application development guidelines to keep the ecosystem vibrant. From this body of knowledge, the process was proposed with eight activities: Specify the Platform, Define MSECO Guidelines, Provide Interface Guidelines, Provide Development Tools, Provide App Management Site, Define App Certification Criteria, Create Incentive Policies, and Provide Apps’ Repository. Since such activities were revised through a peer review study, we will detail them later after presenting the revised MSECO governance process later.

In turn, a peer review with MSECO managers was performed to analyze the proposed governance process. It included an evaluation on how governance strategies have been held by repositories of mobile applications, projects and supporting material, from the perspective of professionals from real ecosystem platforms with experience in MSECO management. The set of questions for the peer review study were defined as follows: 1) A description of the activity *X* was defined correctly and described clearly in the MSECO governance process; 2) The following activity is consistent; 3) The participants of the activity *X* are correct; 4) The names of the artifacts involved in activity *X* are clear; 5) All consumed and produced artifacts are consistent with activity *X*; 6) Activity *X* belongs to a governance process; and 7) Is activity *X* useful?

Initially, characterization forms were prepared to better define the experience of each participant, labeled as KT1 and KT2 (Table 1). The participants with the professional experience related to MSECO management were identified from LinkedIn. We created two documents to support the evaluation: 1) Review Document – contains the sequence of activities, descriptions and artifacts for the participant check if the

Table 1 Participants' profile – peer review

ID	MSECOs	Experience	Profile
KT1	Windows (Microsoft), Symbian (Nokia), Meego (Nokia), and Android (Google).	10 years	High level of experience with mobile application project management, developer community, evangelist team, and partners (startups and companies). They have experience in developer marketing industry projects as well as high expertise in mobile application marketing and management of incentive policies.
KT2	Windows (Microsoft), Android (Google), iOS (Apple), Symbian (Nokia), Meego (Nokia), Gear (Samsung) and Tizen (Samsung)	5 years	

activities are (a) part of a common mobile application development, (b) useful, and (c) have detailed descriptions; and 2) Review Worksheet – aims to support the participant during the evaluation procedure.

In the analysis of “Define MSECO Guidelines” activity, KT1 mentioned that it is missing at least one review step performed by an organizational technology strategy team. For “Create Incentive Policies”, KT1 mentioned that this activity could be disregarded even being observed in the three main MSECO (Android, iOS and Windows). On the other hand, KT2 pointed out that “Specify the Platform” should clearly request the specification of technologies and tools. KT2 also included a suggestion to add an artifact containing app store acceptance criteria in this activity. For “Provide Interface Guidelines”, KT2 missed an explicit list of some ‘reference’ mobile applications (good cases).

In the activity “Provide Development Tools”, KT2 suggested that UX (User Experience) team should be involved in related tasks. KT2 also mentioned that mobile application models to support those who will use the tool should be provided. Regarding “Provide App Management Site”, KT2 indicated that this activity shares goals with “Provide Apps’ Repository”, except the fact that the site where a mobile application undergoes is not the Apps’ Store repository where someone can purchase it (KT1 suggested the inclusion of marketing team as a related role). Finally, KT2 suggested that “Create Incentive

Policies” should be carried out before “Define MSECO Guidelines” since incentive policies can only be set from MSECO guidelines. The inclusion of the marketing team as a role in “Create Incentive Policies” has also been suggested by KT1.

As observed in the interviews, MSECO governance activities focus on the ecosystem health too. As such, they support reuse mechanisms based on MSECO repositories towards reaching the keystone’s goals and ecosystem evolution. Activities also involve some keystone’s teams to produce related artifacts: software development, design, product marketing, marketing developers, legal, product validation, evangelists, and operating system development. Activities related to the definition of strategies aim to support MSECO operation and reuse, including threats or opportunities that may affect the ecosystem. Table 2 shows the revision of MSECO governance activities based on participants’ suggestions.

After performing the peer review study, we identified two artifacts that can serve as reuse mechanisms (or reuse repositories), as described below:

- *Developer Central*: a site that allows developers submit mobile application, as well as application edition, removal and update. This site should allow access to applications’ reports and users’ reviews (e.g. stars and comments). Related links to platform, tools, documents, support,

Table 2 Revision of MSECO governance activities based on experts’ suggestions

Activity	Revised Element	Before	After	Specialist
Specify the Platform	Role	Organization Development Team.	Organization Development and Technology Strategy teams.	KT1
Define MSECO Guidelines	Role	Developer Marketing Team.	Developer Marketing and Technology Strategy teams.	KT1
Provide Interface Guidelines	Produced Artifact	Interface Guidelines.	Interface Guidelines and Reference List of Apps.	KT2
Provide Development Tool	Role	Organization Development Team.	Development and User Experience Teams.	KT2
Provide Development Tool	Required Artifact	Platform Specification and Interface Guidelines.	Reference List of Applications, Platform Specification, and Interface Guidelines.	KT2
Provide App Management Site	Role	Organization Development Team.	Development and Product Marketing Teams.	KT2
Define App Certification Criteria	Role	Organization Product Validation Team.	Product Validation and Development Teams.	KT2
Create Incentive Polices	Role	Evangelists and Legal Team.	Evangelists, Marketing and Legal Teams.	KT2

forums, wikis, and control & publishing information are relevant to this artifact as well; and

- *Apps’ Store*: an environment in which an external developer delivers his/her mobile applications; a user acquires mobile applications; and an ecosystem promotes a democratic place for applications.

The revised version of the proposed MSECO governance process is shown in Fig. 2. Activities are the basis for a well-functioning MSECO, and the keystone is the element responsible for those activities. Activities also involve organizational teams that may contribute to the production and reuse of artifacts as follows: software development, design, product marketing, marketing developers, legal team, product validation, evangelists, and operating system development. The documents generated at the end form the basis of a specific MSECO (artifact: MSECO Guidelines). Such documents are: Platform Specification, Interface Guidelines, Marketing Guidelines, Development Tool, Developer Central, Apps’ Store, Apps’ Store Criteria, MSECO Guidelines, and Incentive Policies. Based on the revised version, we conducted a set of interviews to evaluate the proposed MSECO governance process (Section 4.2).

4.2 Interview with MSECO Managers

To refine the revised MSECO governance process, we conducted semi-structured interviews with 18 managers with professional experience in MSECO management identified from LinkedIn. We contacted a total of 60 managers identified in LinkedIn – 18 answered our survey, giving 85% of confidence level according to the Hamburg’s formula (Hamburg 1980) and a response rate of 30%. The applied data collection strategy was ‘probability sampling’ aiming to eliminate subjectivity and obtain a sample that is both unbiased and representative of the target population. All of them had/have worked in at least one of the following ecosystems: Blackberry, Google Android, iOS, Symbian, watchOS, and Windows. Participants (Table 3) also work/worked in subsidiaries of those organizations in Brazil, Canada, China, Israel, Mexico, and USA. They had an average of 6 (± 3.06) years of professional experience in MSECO management. We obtained a total of 18 suggestions to MSECO governance that guided us to refine the proposed process. To do so, we only used answers that indicated some comment for refinement. The suggestions are described next, as well as the actions taken to refine the approach. The suggestions were classified into: Process, Activities, and Goals.

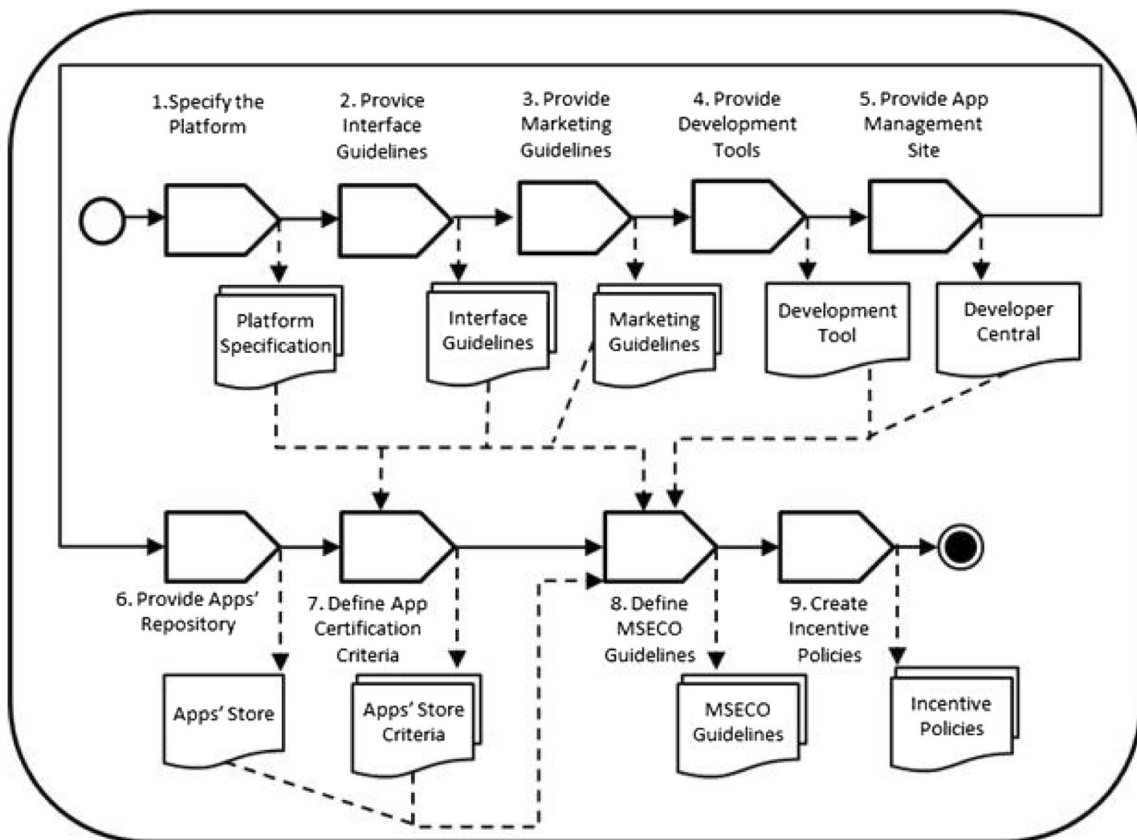


Fig. 2 MSECO Governance Process – revised version

Table 3 Participants' profile – interviews

ID	Ecosystems	Years	Country
1	Android	4	Brazil
2	Android, iOS	2	Canada
3	Windows	4	Brazil
4	Windows	2	Mexico
5	Android, Windows	6	USA
6	iOS	2	Brazil
7	Android	2	Brazil
8	Android, iOS, Windows	3	Brazil
9	Android, Windows, Symbian, S40	5	Brazil
10	Android, iOS, Windows	7	Mexico
11	Windows	5	Canada
12	Windows, BlackBerry	3	Brazil
13	Android	4	Brazil
14	Android	9	China
15	Android, iOS, Windows	10	USA
16	Android, Windows, Nokia	13	Mexico
17	iOS	4	USA
18	Android, iOS, Web	11	Israel

Regarding the *process*, we identified eight potential points of refinement according to the participants' suggestions:

- “*Somehow [the process] evidences the costs and risks that are cited, and how the approach helps mitigate this.*” [P01]

Action: We agree with the participant, because there are software projects that present a set of uncertainties (in different degrees) that can cause problems within an MSECO. To do so, we added the production of an artifact called “Risk Management Plan” in all activities. This artifact consists of risks' identification, analysis, planning, tracking and communication as well as how to mitigate them.

- “*It should consider developer segmentation.*” [P04]

Action: Developer segmentation consists of identifying the niches an MSECO supports. To do so, we added a goal to Activity 8: to identify communities of developers to help the keystone to adapt its strategies and maintain repositories according to niche needs.

- “*I believe the author's intent is clear and well communicated, and I strongly agree with the proposed process. I have minor feedback on the absence of a clear link between “contributing”, “attending” and “recognizing” in a social project management and in a code sharing platform*

like Github. I believe that social platforms as a part of the proposed process should be a strong gain in the value proposition. As an example of a developer environment that does it well, I would like to cite the migration of portions of Unity engine to official Github projects towards leveraging the access to source and community contributions.” [P05]

Action: We accepted the participant's suggestion and included an activity related to the availability and use of repositories that favor interactions among developers.

- “*Overall my guess is that the presented approach makes sense as a starting point.*” [P06]

Action: We agreed with the participant that one can understand that the proposed process could be used only once (to start an MSECO). To do so, we created an activity related to feedback to the process. This feedback can be collected from information related to interactions and contributions of MSECO developers based on repositories.

- “*Developer Marketing work together with Developer Relations, helping improve awareness of content, providing market research, supporting developer events, and creating consistent branding.*” [P09]

Action: We have updated the list of roles involved in the MSECO governance process to include “Developer Relations” team and to consider interactions with “Marketing” team.

- “*In my company there is an area called Developer Relations (DevRel) which includes technology evangelism, community advocacy management, training and documentation. DevRel is working with existing devs and evangelism is getting the word out. Build trust by helping other departments.*” [P12]

Action: We updated the list of roles involved in the MSECO governance process to include “Developer Relations” team joining evangelists.

- “*The process must cover awareness, evaluation, buy in/build, publish, referral/advocacy*” [P14] and “*Relationships are two-way commitments and represent a time investment from the influencing party. Leading and strategic partnerships. Recognition: developers build recognition and achieve their own goals, leading to positive interactions and closer relationships.*” [P10]

Action: Awareness covers the planning and execution of strategies so that developers know the platform and keep

updated. *Evaluation, buy in/build and publish* are covered by Activity 8. *Referral/Advocacy* indicated that an activity for creating/maintaining a Developer Relations team should be added into the process. An activity for seeking developers who can become community leaders is necessary as well.

- “*Developer Marketing involves: demand generation, customer lifecycle, social campaigns, content calendar, email campaigns, brand marketing, digital presence, contests, events, SEO... Developer Relations: it’s a new hybrid role. This is a software engineer who is also gregarious, outgoing and great at public speaking while also writing flawless sample code [...] he/she is active in online communities and has great reputation... To consider: developer evangelism and developer advocacy.*” [P15]

Action: We updated information on the responsibilities of Marketing teams regarding Developers and Developer Relations.

Regarding the *activities*, we identified four potential points of refinement according to the participants’ suggestions:

- “*To demonstrate the temporality of these activities.*” [P02]

Action: We have updated in the process description to inform that activities should be performed more than once as a way to collect feedback and evolve MSECO governance strategies.

- “*A ‘funel’ presents pretty much what real life is.*” [P03]

Action: the participant meant that a mass of developers was initially reached in a “breath” strategy (i.e. hackathon or social media). However, during the engagement tasks within an MSECO, the amount of developers decreases as they specialize or give up of playing in the ecosystem. This fact leads us to think that activities are more related to awareness and onboarding rather than focused on contribution and recognition, for example.

- “*My only concern is about the communication of this schema to the developer’s network and how the conflict management between mainly developers should happen. Despite this point, I think it’s a good proposal.*” [P07]

Action: The process activities produce artifacts that make the communication with developer community easier. In addition, communication is supported by the availability of artifacts in the MSECO repositories. Thus, the organization can explore related information to evolve the process since the developers use the repositories.

- “*I missed the following aspects being addressed in the activities in a clearer way: i. Awareness - awareness regarding the platform and what it does; ii. Acquisition - sign-up/download/install; iii. Activation - actively using the platform through using an application; iv. Retention - continue to use the platform and new/additional features (and in new apps); v. Revenue - pay to use the platform; vi. Referral - tell others about the platform; and vii. Product - involvement in building and getting feedback on product.*” [P13]

Action: We understand that the aspects are already covered by the existing activities. However, specifically in Activity 9 (“Create incentive policies”), aspects related to *retention* and *revenue* aspects can become clearer. For *referral*, we created an activity for team establishment/management. The *product* aspect is considered when collecting feedback on the existing contributions in the MSECO from repositories.

Regarding the *goals*, we identified six potential points of refinement according to the participants’ suggestions:

- “*I would like to see the benefits of the organizational goals for the organization and for the developers (more clearly). Just stating more clearly what those goals represent for the organization in terms of gain (users, revenues, visibility etc.).*” [P01]

Action: We updated activities’ descriptions to represent the benefits indicated by the participant.

- “*I would include some goal related to sales; after all that is the purpose of the organization.*” [P02].

Action: We updated the text related to the process’ objectives in order to cover this suggestion. In addition, we verified whether the sales’ goals were involved in it.

- “*Pretty much complete, just the loop is not clear.*” [P03]

Action: We created an activity that allows the process feedback control through the use of repositories. In addition, we made it clear that each activity also covers monitoring.

- “*I think they are good enough too. It’s very important to keep in mind that these relations must be a win-win model and, for developers, what’s the real impact of his/her work and an expectative to be rewarded (at the same level).*” [P07]

Action: This suggestion became a recommendation of the Activity 9.

- “My suggestion that the process goals must cover the following areas: (1) *Technology Awareness*: spread the word out, generate leads for marketing, pure technological public relations; (2) *Technology Education*: Make developers love your tech, breed community advocates, ease the load on support; and (3) *Technology Feedback*: report to sales, report to marketing, report to product.” [P12]

Action: We updated the texts from the process’ objectives and roles.

- “Developer evangelists may focus much more on activities that result in acquisition, such as docs, blog posts, event sponsorship, and talks. If a company prioritises gathering

product feedback from developers, then advocacy may be the better approach. Developer advocates may undertake all the above activities as part of their developer relations programme.” [P13]

Action: This goal was associated with a new activity related to ‘building and training evangelism and advocacy teams’.

From the interviews, we produced a refined version (Fig. 3) of the proposed MSECO governance process. To do so, we have made updated in *roles, teams, and activities*. The role responsible for the proposed process is:

Role:	Keystone
Description:	Responsible for provisioning standards/practices and identifying opportunities.

The roles involved in the proposed process are:

Role:	Development Team
Description:	Responsible for developing the MSECO platform.
Role:	Design and UX (User eXperience) Team
Description:	Responsible for defining MSECO platform interface standards.
Role:	Product Marketing Team
Description:	Responsible for announcing the MSECO platform and products built on top of it.
Role:	Developer Marketing Team
Description:	Responsible for announcing MSECO developers and promoting their contributions.
Role:	Technology Strategy Team
Description:	Responsible for proposing, defining and analyzing technologies to the keystone.
Role:	Product Validation Team
Description:	Responsible for defining quality requirements for contribution acceptance in the platform.
Role:	Legal Team
Description:	Responsible for caring of the legal part regarding the MSECO elements.
Role:	Developer Relations Team
Description:	Responsible for caring of the liaison between keystone and developers. It is part of an experts’ community within the keystone. Developer evangelists may focus much more on activities that result in acquisitions, such as docs, blog posts, event sponsorship, and talks. If the keystone prioritizes gathering product feedback from developers, then advocacy may be the better approach. Developer advocates may undertake all the above activities as part of their developer relations programme.

The artifacts produced in the proposed process are:

Artifact:	Platform Specification
Description:	Describes how the MSECO platform is organized as well as its level of openness to external developers. In an MSECO, this artifact should describe the allowed mobile devices and their characteristics, programming language, supporting technologies, APIs, SDKs, and the market position (especially to retain users).
Artifact:	Interface Guidelines
Description:	Describes user interface patterns based on mobile devices allowed to communicate with the MSECO platform as well as application-user interactions (e.g. screen patterns and visual elements, components, animations, interaction gestures, messages).

(continued)

Artifact:	Marketing Guidelines
Description:	Describes how to optimize access to in-store applications and how to list tools and guidelines for disclosure soon after publication.
Artifact:	Development Tool
Description:	Allows the application development based on MSECO platform languages, APIs and SDKs. It also allows developers create application binary and package that can be made available in the apps’ store.
Artifact:	Developer Central
Description:	Enables access to platform-related links, tools, documents, support, forums, wikis, and apps’ store’s publication control.
Artifact:	Apps’ Store
Description:	It is a democratic environment for publishing and acquiring MSECO applications.
Artifact:	Apps’ Store Criteria
Description:	Describes the acceptance criteria for quality assessment of MSECO applications. These criteria are based on functional requirements.
Artifact:	MSECO Guidelines
Description:	Synthesizes and aligns all other artifacts generated in the MSECO governance process through a coherent set of guidelines to developers play within the ecosystem.
Artifact:	Incentive policies
Description:	Describes a set of policies to make the strategies defined by the keystone feasible in order to motivate and engage developers to contribute to the MSECO based on keystone’s goals (e.g., publishing application or sharing content/knowledge).
Artifact:	Feedback about MSECO
Description:	Describes information about developers (including their interactions), artifacts (e.g. apps, code template) and technical questions and answers. This report helps the keystone evolve and tailor its governance strategies.

Activities related to the definition of strategies aim to support MSECO operation and reuse processes, considering threats or opportunities that may affect the ecosystem. As the “Platform Specification” artifact is input to all activities, for visualization purposes, we do not place the link as inputs to all activities. All activities should be analyzed for identifying possible risks (and how to mitigate them). To do so, a keystone counts with evangelists, advocates and other

organizational sectors. The following activities compose the proposed process as shown in Fig. 3:

1. **Specify the platform:** supports details on the common technological platform. The goal is to describe the mobile devices supported by the MSECO platform as well as their characteristics, technologies, development languages, APIs, SDKs, and market position.

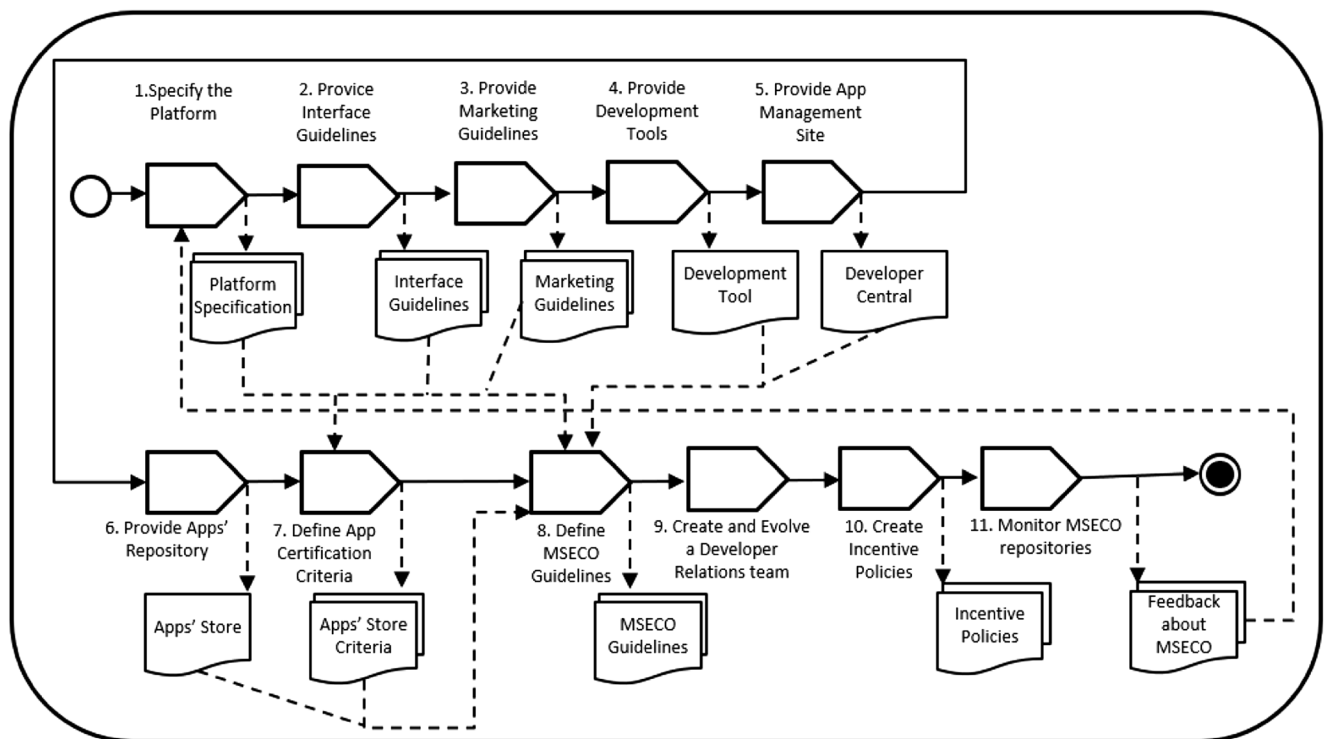


Fig. 3 MSECO Governance Process – refined version

- a. *Participants*: Development Team, Technology Strategy Team;
 - b. *Required Artifacts*: not applicable;
 - c. *Produced artifacts*: Platform Specification;
 - d. *Recommendation*: The specification must have versions for keystone's internal access (more restricted information) and external developers' access in order to not expose confidential information.
2. **Provide Interface Guidelines**: supports user interface as the MSECO platform identity. It is important to publish guidelines to help developers create/develop solutions that meet design standards and platform interface. This activity aims to maintain good user interaction and platform standards.
 - a. *Participants*: Design and UX (User eXperience) Team;
 - b. *Required Artifacts*: Platform Specification;
 - c. *Produced Artifacts*: Interface Guidelines;
 - d. *Recommendation*: Considerations on the platform's limitations and potentials should be done in this activity.
 3. **Provide Marketing Guidelines**: supports mobile application (and developer) visibility. A set of tabs should be created to quickly deliver applications to the market. In addition, there should be tools to foster application's and developer's popularities.
 - a. *Participants*: Product Marketing Team, Developer Marketing Team;
 - b. *Required Artifacts*: Platform Specification, Interface Guidelines;
 - c. *Produced Artifacts*: Marketing Guidelines;
 - d. *Recommendation*: The following marketing guidelines should be considered in this activity: marketing within the Apps' Store; marketing outside the Apps' Store; and marketing within the supported mobile devices.
 4. **Provide Development Tools**: supports tools for building applications on top of the MSECO platform.
 - a. *Participants*: Development Team, Design and UX (User eXperience) Team;
 - b. *Required Artifacts*: Platform Specification;
 - c. *Produced Artifacts*: Development Tool;
 - d. *Recommendation*: A development tool should allow the following tasks: access to platform guides; development of a complete application; testing planning and execution, at least at the unit level; application code debugging; packet generation in debug and delivery modes; interaction with platform device emulators.
 5. **Provide App Management Site**: supports a website for application management and delivery.
 - a. *Participants*: Development Team, Product Marketing Team;
 - b. *Required Artifacts*: Platform Specification;
 - c. *Produced Artifacts*: Developer Central;
 - d. *Recommendation*: At that time, a developer is a content publisher and MSECO repositories should contain all the information and/or links that can help his/her during the application submission and management tasks.
 6. **Provide Apps' Repository**: supports a distribution channel for developers (application delivery), for users (application acquisition), and for the ecosystem (democratic environment that supports the social network). There should be an application management portal based on the MSECO platform specification and interface design guidelines.
 - a. *Participants*: Development Team;
 - b. *Required Artifacts*: not applicable;
 - c. *Produced Artifacts*: Apps' Store;
 - d. *Recommendation*: The environment must be democratic from the following points of view:
 - i. *Developer*: it should support participation of many types of developers (individuals or companies), offering options to make an application available for free or paid;
 - ii. *User*: it should support a store that allows application search, recommendation, acquisition, and reviewing. The user can purchase free/paid apps and apps in testing.
 7. **Define App Certification Criteria**: supports the definition of quality assurance criteria for mobile applications to ensure they meet platform and user interface requirements.
 - a. *Participants*: Product Validation Team, Development Team;
 - b. *Required Artifacts*: Platform Specification, Interface Guidelines, Marketing Guidelines;
 - c. *Produced Artifacts*: Apps' Store Criteria;
 - d. *Recommendation*: Certification criteria should consider mobile devices supported by the MSECO platform, including hardware/software specification, interface design guidelines, and countries' legislations for digital content access issues.
 8. **Define MSECO Guidelines**: supports the set of artifacts produced in previous activities to create a single document. Such document should be available at the MSECO portal or website (quick and easy

access). Keystone must work on identifying developers who belong to ecosystem communities. This will favor the planning of strategies that more effectively affect developers' engagement.

- a. *Participants*: Developer Marketing Team, Technology Strategy Team;
 - b. *Required Artifacts*: Platform Specification, Interface Guidelines, Marketing Guidelines, Development Tool, Developer Central, Apps' Store, Apps' Store Criteria;
 - c. *Produced Artifacts*: MSECO Guidelines;
 - d. *Recommendation*: Guidelines should make a coherent link among all artifacts generated in previous activities and they should be easily accessible for any potential external developer.
9. **Create and Evolve a Developer Relations team**: supports the keystone as a team of professionals with technical experience and good communication with the developer community. This team aims to strengthen the organization's relationship with the MSECO developer community.
- a. *Participants*: Developer Relations Team;
 - b. *Required Artifacts*: MSECO Guidelines;
 - c. *Produced Artifacts*: not applicable;
 - d. *Recommendation*: The purpose of the keystone is to leverage sales. Therefore, this team must include experts in business, technical and social dimensions of an MSECO.
10. **Create Incentive Policies**: supports definition, planning and execution of MSECO strategies, e.g. how to ensure developer motivation and recognition, a productive ecosystem, and stable platform architecture. This activity aims to define policies that encourage developer participation/engagement within a specific MSECO;
- a. *Participants*: Developer Relations Team, Legal Team, Developer Marketing Team;
 - b. *Required Artifacts*: MSECO Guidelines;
 - c. *Produced Artifacts*: Incentive Policies;
 - d. *Recommendation*: Policies should recognize contribution and developer, support developer engagement, define rules for stakeholders including those responsible for developer support (evangelists), and be validated by the legal organization based on regulation mechanisms. It is very important to keep in mind that these relations must be driven to a win-win model and to developers, informing the real impact of their work and possible rewards.
11. **Monitor MSECO repositories**: supports the keystone to collect repository indicators regarding artifact usage information and developer interactions. Developer

Relations team can do it and communicate to the other organizational teams.

- a. *Participants*: Developer Relations Team;
- b. *Required Artifacts*: MSECO Repositories (Developer Central and Apps' Store);
- c. *Produced Artifacts*: Feedback about MSECO;
- d. *Recommendation*: Information collected from MSECO repositories serves for ecosystem monitoring and also for providing the keystone with feedback on the MSECO products/tools.

After presenting the refined MSECO governance process based on interviews, we describe an exploratory study to analyze how the MSECO content reuse repositories identified from the previous studies can contribute to the process as well as and how they can be monitored in the next section.

5 Exploring Content Reuse Repositories to Support MSECO Governance Process

This section presents an exploratory study planned and executed with the goal of **analyze** the structure of the MSECO content reuse repositories **in order to** characterize **in relation to** their common characteristics **from the point of view of** researchers **in the context** of Apple, Google and Microsoft MSECO. The analysis of MSECO content reuse repositories' structure can help us to know their common characteristics regarding, for example, how ecosystems compete, comparison between governance strategies and effects on applications or developer communities. As such, a keystone can adapt governance strategies to keep the ecosystem vibrant as well as to attract developers and users (Axelsson and Skoglund 2016).

In this context, we performed an analysis of those repositories indicated by MSECO managers in the study presented in Section 4. Based on this study, we observed that the repositories used in the main existing MSECO in industry (Fontão et al. 2015) are: mobile applications (App Store from Apple,² Google Play from Google,³ and Windows Store from Microsoft⁴), support material (Apple Developer,⁵ Android Developers,⁶ and Microsoft Developers⁷) and mobile applications management (from each platform).

In the **first step**, a software engineering researcher (MSECO expert) accessed each repository and set up the repositories' structure through a mind map. Next, in the **second**

² <https://itunes.apple.com/us/genre/ios>

³ <https://play.google.com/store?hl=en>

⁴ <https://www.microsoft.com/en-us/store/apps/windows-phone>

⁵ <https://developer.apple.com>

⁶ <https://developer.android.com>

⁷ <https://developer.windows.com>

step, common elements were entered into a new mind map to produce a MSECO content reuse repositories' common structure. Since the previous steps were performed by the same researcher, a further step was executed to reduce the bias of researcher subjectivity and failure in the construction of the common structure. As such, in the **third step**, four researchers (MSECO experts) evaluated the built mind map. After these steps, those five researchers (Table 4) produced the final mind map (Fig. 4) with the MSECO content reuse repositories' common structure.

After analyzing the selected repositories, it was revealed that a governance process artifact called “Developer Central” (Fig. 3) was split into two artifacts: “Developer Central” and “Apps’ Management”. In the following sections, we discuss the characteristics of such three repositories.

5.1 Apps’ Store

Mobile applications should be approved by the MSECO platform's acceptance criteria defined by the keystone. After that, they can become available so that users can download them. In this regards, Apps’ Store is an MSECO content reuse repository that comprises information on applications and developers, but focuses mainly on users. Apps’ Store lets developers easily deliver applications to users over the world as well as integrate advanced features. It is composed by: Categories (e.g. Entertainment, Health & Fitness, Education, Finance), Featured (slots to enhance visibility of selected applications), Top Paid Apps/Games, Top Free Apps/Games, New Releases (new applications available at Apps’ Store), Updates (user notifications on applications), and Collections (e.g. Christmas, Football Lovers, and Winter).

Apps’ Store works as a physical product store, where the window (“balcony”) is organized as described above. The keystone designs the catalog exhibition to attract users and engage those who have already downloaded and evaluated mobile applications. As such, users have an active role in the store, because they evaluate mobile applications with comments/stars after acquired them. Comments/stars serve as information for keystone and developers to better meet user demands within the ecosystem.

5.2 Developer Central

Developer Central is an MSECO content reuse repository that provides and hosts materials to support external developers in their activities. The artifact “MSECO Guidelines” are available to developers through this repository. Based on the MSECO content reuse repositories' common structure, Developer Central categorizes materials related to the sections (that comprised different repository elements of functions as describe next): Platforms, Design, Development, Distribution, and Support.

In *Platforms* section, materials are related to existing opportunities within the ecosystem, information regarding platform features, interface and APIs, new resources, and videos highlighting platform strengths. In *Design*, resources are related to teach how to design and code interfaces for mobile devices supported by the MSECO platform. Material available to the external developer supports the *typography* (visual representation of the application language), *usability* (tips for a good user experience), *accessibility* (accessible applications for people who have any limitation), and *adaptability* (application content should be displayed correctly in different mobile devices). In order to support interface design and construction, reusable components can be provided as well as instructional videos that help developers to learn about them. All mentioned MSECO repositories offer packages with code templates and documentation.

In *Development*, there are ‘how-to’ articles, instructions and code examples for all tasks, such as transferring and consuming data over a network, geolocation services, and navigation between screens. In this regards, a keystone wants to stimulate developers to bring their ideas to life, creating mobile applications to expand the ecosystem. In the Developer Central, the keystone provides tools to support application development (e.g. XCode, Android Studio, and Visual Studio) with free and paid versions, patterns that should be adopted in the platform development, and standards for assessing application performance. In training, developers will find classes on how to accomplish a specific task with code samples he/she can reuse in his/her application. Classes are organized into several groups.

Table 4 Participants’ profile – exploratory study

ID	Ecosystems	Experience Years (Research)	Experience Years (Industry)	Profile
R1	PhD	14	10	Software Engineering, Software Ecosystem and Software Reuse
R2	PhD	14	6	Software Engineering, Software Testing, Empirical Software Engineering
R3	PhD Candidate	5	7	Software Engineering, Mobile Software Ecosystem, Developer Governance
R4	PhD Candidate	6	4	Software Engineering, Software Testing, Empirical Software Engineering
R5	PhD Candidate	5	4	Software Engineering, Empirical Software Engineering

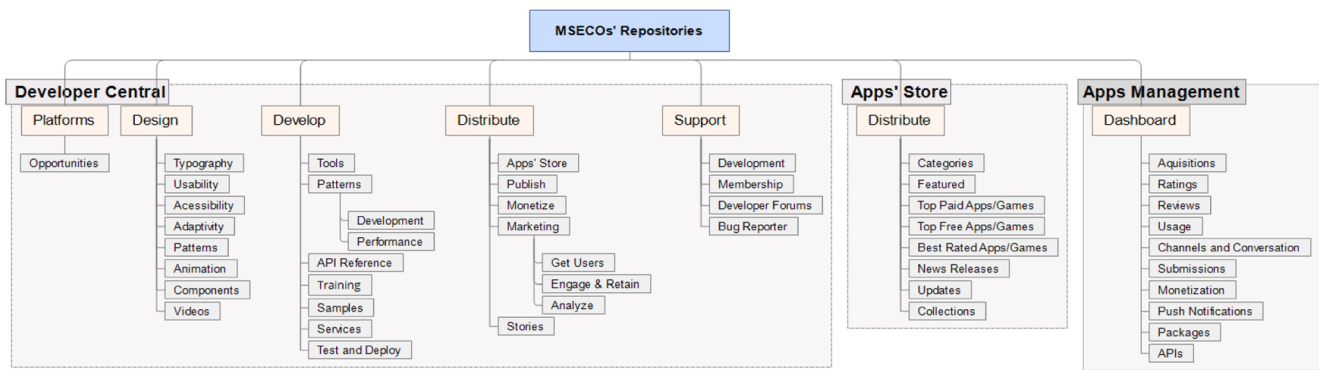


Fig. 4 The final mind map produced by researchers after analyzing MSECO content reuse repositories

Also regarding *Development*, application program interface (API) is a set of routines and protocols that specify how components should interact. There are sample codes available in Developer Central section, including a category-based section in which contributors can browse sample code and learn how to build different components for his/her applications. Such code samples show simple solutions for common problems and simple recipes to help developers implement new mobile application features. For example, user interface sample codes contain demos, experiments and prototypes. There is a collection of ‘helper’ functions, custom components and application services created by MSECO developers, including documentation and packages.

Extra tasks are also covered by *Development*. ‘Services’ provide an easy and secure way to perform a set of activities, e.g. to pay for physical goods and services such as groceries, clothing, and tickets through mobile applications. Other examples are services that allow developers to read/write health and activity data to their health-domain applications. As such, during application development, it is also necessary to ensure that a test & deploy section is available to help developers in debugging and testing activities, describing how to improve performance and use tools (e.g. emulators/devices), APIs (e.g. Junit), techniques (e.g. unit tests).

Distribution section refers to all resources that help developers in publishing their mobile applications at an Apps’ Store. ‘Publishment’ supports developers to prepare their application submission including: guidance for enrolling in a publisher program, configuration services available only for applications submitted to the Apps’ Store, procedure to upload application metadata and support user search/retrieval, and tips on how to release and maintain an application after submission. When a developer requires resources to make money with his/her applications, ‘Monetize’ describes potential strategies and offers a variety of tools to support monetization. Strategies includes: increase revenue by using services to display banner ads and video interstitial ads; offer products and features that users can buy only from an application; and limit features in a free trial version to induce users to buy an application.

Moreover, ‘Marketing’ section comprises how to ‘Get Users’, ‘Engage & Retain’, and ‘Analyze’ them in their interaction with mobile applications. Resources available in ‘Get Users’ refers to attract users, i.e. best practices for getting successful applications (e.g. use ads, improve marketing, expand to new markets, and understand where users come from). In turn, it is necessary to ‘Engage & Retain’ users. For example, Android Developers section mentions that keeping active users is the key for a successful application. As such, several tools and techniques to keep users coming back to the application are suggested: build useful widgets, engage users based on their language, notify users to keep them informed, increase usage with easy search, and encourage user competition. Finally, ‘Analyze’ the users’ engagement refers to invite users to tell what they think about an application as well as to follow up and connect with user by responding their reviews either publicly or privately. ‘Stories’ has articles and/or videos where developers share experience over the MSECO platform and Apps’ Store. Developers share how they get successful over the Apps’ Store and what they have learned along the way.

Finally, another important common section is *Support*. It refers to a technical advice channel personalized with enrollment, membership, development, and tools. In this section, ‘Developer Forums’ consist of a social environment where developers can post questions and share thoughts with fellow developers and evangelists. Support also allows developers play as ‘Bug Reporters’, informing bugs and requesting enhancements to APIs and developer tools.

5.3 Apps’ Management

Apps’ Management refers to an MSECO content reuse repository that allows developers to submit and manage their applications. This repository differs from Apps’ Store because Apps’ Management also keeps mobile applications that were not published at the store. Some elements compose such repository: Acquisitions, Ratings, Reviews, Usage, Channels and Conversation, Monetization, Push Notifications, and Submissions/Packages. These elements are described below:

- *Acquisitions*: it allows developers to see who purchased their applications, along with demographic and platform details. Data can also be visualized in the online dashboard. In a download report, acquisition means that a user has obtained a license for a developer's application;
- *Ratings*: allows developers to see distribution of how many users' ratings their mobile applications have. In this report, classification means the number of stars (1 to 5) that a user gave to a developer's application in the Apps' Store. This report does not include information on all individual comments left in the Apps' Store as reviews. It is possible to apply filters (by date, market niche and device);
- *Reviews*: it allows developers to see user comments on their mobile applications in the store. Each user opinion contains: title and text (opinion) provided by the user; review date; name, user country/ region; application package version; user operating system; and device name;
- *Usage*: it allows developers to see how many people are using their mobile applications. Each user session is a distinct period when a user interacted with an application. Developers can analyze the number of users who used the application (and screens) in a given period of time. This section also helps to understand user behavior and user value over time. Developers can explore what makes different groups of users unique and how such groups respond to application content, features, and monetization strategies;
- *Channels and Conversion*: a channel refers to how a user finds a mobile application (e.g. navigation and search over Apps' Stores, links to external site, or links to custom campaigns). A page view means that a user visited an App's Store listing page (Web-based or Mobile-based). In turn, a conversion means that a user recently obtained a license for an application (either paid or free). The following types of channel are included: Traffic store – a user was browsing or searching the store when visualized application details; External site – a user followed a link (without the custom campaign ID) to access application details at a website; Search engine – a user followed a link to application details that was returned by an online search engine; and Custom campaign – a user followed a link that has the custom campaign ID;
- *Monetization*: it is used to access control settings and add parameters required for each advertising network/purchase used by an application. Advertising control allows developers to optimize revenue;
- *Push Notifications*: it engages users at the right time and with the right message as a success factor for a mobile application. Apps' Management provides a data-driven user engagement platform a developer can use to send out push notifications to all users (or only those who meet any [user segment](#) criteria). Developers can use push

notifications to encourage users to take an action, such as rating an application, buying an add-on, trying a new feature, or downloading another application;

- *Submissions/Packages*: refers to application information used for all MSECO platforms. It contains information about Name, Privacy Policy URL, Primary Language, Screenshots, Description, Keywords, Rating, License Agreement, Pricing, and Availability. Packages must contain the following status: In progress; In Store; or Not available in Store.

Our goal with this study was to identify a standard framework among Android MSECO, iOS MSECO, and Windows Phone MSECO. The identification of the common structure favors ecosystem comparison and analysis of competition or how such ecosystems collaborate. An important aspect to be investigated is how ecosystems differ in relation to their internal content reuse repositories and how their characteristics affect the MSECO health as well as its constituent elements. Regarding an awareness strategy based on repositories, the analyzed ecosystems cover the following aspects: ecosystem platforms, interface standards (how contributions can be made at the platform), the best contribution distribution strategy, and marketing and technical support for developers' community (i.e. individuals, partners companies, and startups). Regarding the Apps' Store, further studies are required to evaluate the window display differentiation for each MSECO and how it attracts users and allows developers to address impact information from other applications. Discussion surrounding the store also involves whether it could be considered as an information reuse repository from the developers' perspective. Our analysis along with the previous studies presented in this paper allowed us to realize that the migration of functionalities between different stores or MSECO platforms can support a reuse strategy. In this context, Apps' Stores can be useful in recommending what features a developer could implement, and what strategy can best suit a particular developer's landscape so that he/she can achieve success in the ecosystem.

In this work, we analyzed the main internal content reuse repositories indicated by the participants of the previous studies (peer review and interviews). There are several researches that look at MSECO external repositories, such as StackOverflow (Novielli et al. 2014) (Calefato et al. 2015) (Serva et al. 2016) (Fontão et al. 2017) and Github (Casalnuovo et al. 2015) (Vasilescu et al. 2015) (Constantinou and Mens 2017). As a conclusion, a further analysis should address how the information linked between internal and external content reuse repositories can favor keystone's decisions and developer community engagement.

6 Threats to Validity

In this section, we identify possible threats to the validity of our review results based on the taxonomy of Wohlin et al. (2013).

Constructo validity: the main threats in this category are related to the construction of the MSECO governance process and the questionnaires used for both initial studies (peer review and interviews with the managers). To reduce such threats, we used data from (Fontão et al. 2015) and (Manikas 2016) to build the initial version of the proposed process since both references are systematic mapping studies on MSECO and software ecosystems. Moreover, the proposed process was revised and refined through studies with experts as presented in Section 4. Both questionnaires were elaborated following the guidelines for conducting these types of studies proposed by Wohlin et al. (2013). Finally, in the exploratory study, the materials analyzed by the five researchers consisted of the official repositories of the existing MSECO in industry (Section 5).

Internal validity: for this study, we proposed to select managers who work in the main existing MSECO in the software industry. Thus, we assumed that they are representative for the population of MSECO managers and helped to revise and refine the proposed process from a real scenario perspective.

Conclusion validity: it was accomplished through simple demonstration of presence (or not) of activities, artifacts and roles defined in the proposed MSECO governance process.

External validity: it refers to issues that affect the strength to conclude that the studies can be repeated with the same results. As mentioned in internal validity, participants play in the main existing MSECO in the software industry. However, new rounds of studies could be performed with more MSECO managers. We believe that our study can be replicated following the steps described in Sections 4 and 5.

7 Conclusion and Future Work

In MSECO, central organizations – also known as keystones – need to find mechanisms to manage and monitor external developers and their contributions to the ecosystem. As such, keystones should use governance strategies to attract, engage and retain developers towards a vibrant MSECO. In this paper, we investigated MSECO governance strategies that emerge from content reuse repositories. To do so, we designed and analyzed a MSECO governance process and evaluated it

with managers who work(ed) in the main MSECO organizations to identify the main governance activities, artifacts and roles. Next, we performed interviews with 18 experienced managers to refine the governance process. Then, we analyzed the main MSECO reuse repositories to understand how reuse mechanisms and governance strategies have supported keystone and external developers to contribute to MSECO expansion.

In the proposed process, we identified different forms to manage knowledge and contributions within a specific MSECO. From MSECO content reuse repositories, governance strategies were explored through analyses of Apple Developer, Android Developers, Windows Developer, App Store, Google Play, and Windows Store. To do so, five researchers (MSECO experts) derived a content reuse repositories' common structure describing shared characteristics obtained from those repositories: (1) Developer Central repository joins information on platforms, design, development, distribution, and official support channels; (2) Apps' Store offer applications and work similarly to a physical store, distributing applications by categories, collections, featured, top paid/free applications or games, best rated, news, and updates; and (3) Apps' Management repository supports application projects to be stored, offering usage, ratings, reviews, and download reports. Each content reuse repository can help keystones to know how ecosystems compete, compare governance strategies and be aware of effects on applications or developer communities, as described in Section 5. Based on this exploratory study, we can conclude that ecosystem health indicators can be further investigated in MSECO. Moreover, further studies regarding the MSECO governance process will be conducted with developers and users.

As a contribution, our research concluded that MSECO content reuse repositories are crucial for the ecosystem expansion since reuse must be managed and monitored by the keystone (organization that maintains the MSECO platform). In this regards, artifact quality is an important concern to be explored since it serve as the basis for external developers creating/evolving contributions, e.g. study of bad smells in sample codes. As future work, we intend to investigate how MSECO content reuse repositories can be mined to identify communities, user behavior, quality of contributions, and developer interaction. Another future work refers to explore how social network analysis can help to refine and assess MSECO governance strategies. Finally, developers' experiences with sample codes, tools, MSECO portals and documentation should be studied from the point of view of Developer eXperience (DX). Similarly, user experiences with applications in the Apps' Store can be analyzed from the point of view of User eXperience (UX). In this regards, MSECO external repositories such as GitHub, Codeplex and Stackoverflow can be analyzed together with MSECO content

reuse repositories (internal) in order to identify sided effects and opportunities to expand an MSECO.

Acknowledgements The authors would like to thank FAPEAM, CNPq and CAPES for the financial support. The second author also thank to DPq/PROPGPI/UNIRIO for the partial support to this research.

References

- Ahasanuzzaman, M., Asaduzzaman, M., Roy, C.K., Schneider, K.A. (2016). Mining duplicate questions in stack overflow. Proc. 13th Int. Work. Min. Softw. Repos. - MSR '16 402–412. <https://doi.org/10.1145/2901739.2901770>.
- Alves, C., Oliveira, J., Jansen, S. (2017). Software Ecosystems Governance - A Systematic Literature Review and Research Agenda. Proc. 19th Int. Conf. Enterp. Inf. Syst. 215–226. <https://doi.org/10.5220/0006269402150226>.
- Ascate, S.M., Oliveira, K., Mendes, I., Fontão, A., Villanes, I.K., Dias-Neto, A. (2017). Model-based testing for mobile applications: A systematic mapping study, in: CIBSE 2017 - XX Ibero-American Conference on Software Engineering.
- Axelsson, J., & Skoglund, M. (2016). Quality assurance in software ecosystems: A systematic literature mapping and research agenda. *Journal of Systems and Software*, 114, 69–81. <https://doi.org/10.1016/j.jss.2015.12.020>.
- Baars, A., & Jansen, S. (2012). A framework for software ecosystem governance. In: International Conference of Software Business. Springer, pp. 168–180.
- Barbosa, O., Dos Santos, R., Alves, C., Werner, C., Jansen, S. (2013). A systematic mapping study on software ecosystems, in: Jansen, S., Brinkkemper, S., Cusumano, M.A. (Eds.), *Software ecosystems. Analyzing and managing business networks in the software industry*. Edward Elgar publishing limited, pp. 59–78.
- Bentley, P., Lim, S. (2012). How to be a Successful App Developer: Lessons from the Simulation of an App Ecosystem 129–136. <https://doi.org/10.1145/2384697.2384698>.
- Bosch, J. (2009). From Software Product Lines to Software Ecosystems, in: SPLC '09 Proceedings of the 13th International Software Product Line Conference. Carnegie Mellon University Pittsburgh, PA, USA, pp. 111–119.
- Calefato, F., Lanubile, F., Marasciulo, M.C., Novielli, N. (2015). Mining successful answers in stack overflow. IEEE Int. Work. Conf. Min. Softw. Repos. 2015–August, 430–433. <https://doi.org/10.1109/MSR.2015.56>.
- Casalnuovo, C., Vasilescu, B., Devanbu, P., Filkov, V. (2015). Developer onboarding in github: the role of prior social links and language experience. ESEC/FSE conf. 1–12. <https://doi.org/10.1145/2786805.2786854>.
- Constantinou, E., Mens, T. (2017). Socio-technical evolution of the Ruby ecosystem in GitHub. SANER 2017 - 24th IEEE Int. Conf. Softw. Anal. Evol. Reengineering 34–44. <https://doi.org/10.1109/SANER.2017.7884607>.
- Dhungana, D., Groher, I., Schludermann, E., Biffl, S. (2010). Software Ecosystems vs. Natural Ecosystems : Learning from the Ingenious Mind of Nature. Proc. 4th Eur. Conf. Softw. Archit. Companion Vol. (ECSA '10) 96–102. <https://doi.org/10.1145/1842752.1842777>.
- Eckhardt, E., Kaats, E., Jansen, S., Alves, C. (2014). The merits of a meritocracy in open source software ecosystems, in: Proceedings of the 2014 European Conference on Software Architecture. <https://doi.org/10.1145/2642803.2642810>.
- Fagerholm, F., Ikonen, M., Kettunen, P., Münch, J., Roto, V., & Abrahamsson, P. (2015). Performance alignment work: How software developers experience the continuous adaptation of team performance in lean and agile environments. *Information and Software Technology*, 64, 132–147. <https://doi.org/10.1016/j.infsof.2015.01.010>.
- Fontão, A., Santos, R.P., Dias-neto, A.C. (2015). Mobile Software Ecosystem (MSECO): A Systematic Mapping Study, in: COMPSAC 2015 - The 39th Annual International Computers, Software & Applications Conference. pp. 653–658.
- Fontao, A., Bonifácio, B., Bezerra, A., Dias-Neto, A., Santos, R. (2015a). MSECO-SUP: Support process in mobile software ecosystems, in: Simpósio Brasileiro de Engenharia de Software. <https://doi.org/10.1109/SBES.2015.20>.
- Fontao, A., Pereira, R., Dias-Neto, A. 2015b. Research opportunities for mobile software ecosystems, in: WDES 2015 - Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems. pp. 4–5.
- Fontão, A.L., Santos, R.P., Filho, J.F., Dias-Neto, A.C. (2016). MSECO-DEV: Application development process in mobile software ecosystems. In: Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE. <https://doi.org/10.18293/SEKE2016-157>.
- Fontao, A., Ekwoje, O.M., Dias-Neto, A.C., Santos, R. (2017). Facing up the primary emotions in Mobile Software Ecosystems from Developer Experience, in: ACM International Conference Proceeding Series. <https://doi.org/10.1145/3098322.3098325>.
- Fontão, A., Lima, F., Ábia, B., dos Santos, R.P., Dias-Neto, A.C. (2017). Hearing the Voice of Developers in Mobile Software Ecosystems, in: Proceedings of the 31st Brazilian Symposium on Software Engineering - SBES'17. pp. 4–13. <https://doi.org/10.1145/3131151.3131167>.
- Frakes, W. B., & Fox, C. J. (1996). Quality improvement using a software reuse failure modes model. *IEEE Transactions on Software Engineering*, 22, 274–279.
- Franco-Bedoya, O., Ameller, D., Costal, D., & Franch, X. (2017). Open source software ecosystems: A systematic mapping. *Information and Software Technology*, 91, 160–185. <https://doi.org/10.1016/j.infsof.2017.07.007>.
- Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. *Journal of Systems and Software*, 125, 207–219. <https://doi.org/10.1016/j.jss.2016.11.027>.
- Haenni, N., Lungu, M., Schwarz, N., & Nierstrasz, O. (2014). A quantitative analysis of developer information needs in software ecosystems. in: Proceedings of the 2014 European Conference on Software Architecture Workshops. Article No. 12.
- Hamburg, M., 1980. Basic statistics: A modern approach. J. R. Stat. Soc. 143.
- Hanssen, G. K. (2012). A longitudinal case study of an emerging software ecosystem: Implications for practice and theory. *Journal of Systems and Software*, 85, 1455–1466. <https://doi.org/10.1016/j.jss.2011.04.020>.
- Jansen, S., Brinkkemper, S., & Cusumano, M. A. (2013). *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, Edward Elgar Publishing.
- Jansen, S., Cusumano, M.A. 2012. Defining software ecosystems : A survey of software platforms and business network governance, in: Proceedings of the International Workshop on Software Ecosystems 2012. pp. 41–58.
- Lin, F.L.F., Ye, W.Y.W. 2009. Operating System Battle in the Ecosystem of Smartphone Industry. 2009 Int. Symp. Inf. Eng. Electron. Commer. <https://doi.org/10.1109/IEEC.2009.136>.
- Mäenpää, H., Munezero, M., Fagerholm, F., Mikkonen, T. 2017. The many hats and the broken binoculars, in: Proceedings of the 13th International Symposium on Open Collaboration - OpenSym '17. pp. 1–9. <https://doi.org/10.1145/3125433.3125474>.

- Manikas, K. (2016). Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software*, 117, 84–103. <https://doi.org/10.1016/j.jss.2016.02.003>.
- Manikas, K., Hämäläinen, M., & Tyrväinen, P. (2016). Designing, developing, and implementing software ecosystems: towards a step-wise guide. *CEUR Workshop Proc.*, 1808, 70–79.
- Manikas, K., Wnuk, K., Shollo, A. (2015). Defining decision making strategies in software ecosystem governance. Technical Report - Department of Computer Science, University of Copenhagen.
- Martin, W., Sarro, F., Jia, Y., Zhang, Y., & Harman, M. (2017). A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering*, 43, 817–847. <https://doi.org/10.1109/TSE.2016.2630689>.
- McGregor, J.D., (2010). A method for analyzing software product line ecosystems. Proc. Fourth Eur. Conf. Softw. Archit. Companion Vol. - ECSA '10 73. <https://doi.org/10.1145/1842752.1842773>.
- Messerschmitt, D. G., & Szyperski, C. (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*, MIT Press Books. Cambridge: The MIT Press.
- Neto, V. V. G., Araujo, R., & Dos Santos, R. P. (2017). New challenges in the social web: Towards systems-of-information systems ecosystems. *CEUR Workshop Proc.*, 2039, 1–12.
- Novielli, N., Calefato, F., & Lanubile, F. (2014). Towards discovering the role of emotions in stack overflow. *Proc. 6th Int. Work. Soc. Softw. Eng. - SSE, 2014*, 33–36. <https://doi.org/10.1145/2661685.2661689>.
- Pant, V., Yu, E. (2017). Understanding strategic moves and reciprocity on Software Ecosystems : A strategic modeling approach. pp. 28–42.
- Sadi, M. H., Dai, J., Yu, E. (2015). Designing Software Ecosystems: How to Develop Sustainable Collaborations? In: International Conference on Advanced Information Systems Engineering. Springer, pp. 161–173. https://doi.org/10.1007/978-3-319-19243-7_17.
- Santos, R., & Werner, C. (2012). ReuseECOS: An approach to support global software development through software ecosystems. IEEE Seventh International Conference on Global Software Engineering Workshops. pp. 60–65. <https://doi.org/10.1109/ICGSEW.2012.16>.
- Santos, R.P., Werner, C. (2011). Treating business dimension in Software Ecosystems, in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES'11. pp. 197–201. <https://doi.org/10.1145/2077489.2077526>.
- Santos, R., Lima, T., Werner, C., Tostes, L., & Barbosa, G. (2017). A sociotechnical negotiation mechanism to support component markets in software ecosystems. *CLEI Electron. J.*, 20, 4. <https://doi.org/10.19153/cleiej.20.3.4>.
- Schreieck, M., Wiesche, M., Krcmar, H. (2016). Design and Governance of Platform Ecosystems – Key Concepts and Issues for Future Research. *Ecis* 1–20.
- Serva, R., Senzer, Z.R., Pollock, L., Vijay-Shanker, K. (2016). Automatically mining negative code examples from software developer Q & A forums. Proc. - 2015 30th IEEE/ACM Int. Conf. Autom. Softw. Eng. Work. ASEW 2015 115–122. <https://doi.org/10.1109/ASEW.2015.10>.
- Song, J., Baker, J., Wang, Y., Choi, H. Y., & Bhattacharjee, A. (2018). Platform adoption by mobile application developers: A multimethodological approach. *Decision Support Systems*, 107, 26–39. <https://doi.org/10.1016/j.dss.2017.12.013>.
- Stol, K.-J., & Fitzgerald, B. (2015). Inner source-adopting open source development practices in organizations: A tutorial. *IEEE Software*, 32, 60–67. <https://doi.org/10.1109/MS.2014.77>.
- Valença, G., & Alves, C. (2017). A theory of power in emerging software ecosystems formed by small-to-medium enterprises. *Journal of Systems and Software*, 134, 76–104. <https://doi.org/10.1016/j.jss.2017.08.044>.
- Wasilescu, B., Serebrenik, A., Filkov, V. (2015). A data set for social diversity studies of GitHub teams. IEEE Int. Work. Conf. Min. Softw. Repos. 2015–August, 514–517. <https://doi.org/10.1109/MSR.2015.77>.
- Wareham, J., Fox, P. B., & Cano Giner, J. L. (2014). Technology ecosystem governance. *Organization Science*, 25, 1195–1215. <https://doi.org/10.1287/orsc.2014.0895>.
- Wohlin, C., Runeson, P., Neto, P. A. M. S., Engström, E., Carmo Machado, I., & Almeida, E. S. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, 86, 2594–2610. <https://doi.org/10.1016/j.jss.2013.04.076>.

Awdren de Lima Fontão is PhD student in Informatics at the Institute of Computing, Federal University of Amazonas (UFAM), where he researches Software Ecosystems, Mobile Applications, Developer Relations and related topics. He is also member of the Experimentation and Testing on Software Engineering Group (EXPERTS). Contact him at wdrn@icomp.ufam.edu.br.

Rodrigo Pereira dos Santos is Assistant Professor in the Department of Applied Informatics at the Federal University of the State of Rio de Janeiro (UNIRIO), where he researches Software Ecosystems, Requirements Engineering and topics related to Software Engineering as a member of the Graduate Program in Informatics. He is Editor-in-chief of iSys: Brazilian Journal of Information Systems. Contact him at rps@uniriotec.br.

Arilo Claudio Dias-Neto is Associate Professor in the Institute of Computing at the Federal University of Amazonas (UFAM). He also leads the Experimentation and Testing on Software Engineering Group (EXPERTS) and participates in industry R&D projects. Contact him at arilo@icomp.ufam.edu.br.