Health Measurement of Data-Scarce Software Ecosystems: A Case Study of Apple's ResearchKit

Paul van Vulpen(✉), Abel Menkveld, and Slinger Jansen

Utrecht University, Utrecht, The Netherlands
{p.n.vanvulpen,a.a.menkveld}@students.uu.nl,
slinger.jansen@uu.nl

Abstract. Current methods for measuring open source software ecosystem health are unable to measure the health of young software ecosystems, due to a lack of data. This paper proposes a new method for measuring software ecosystem health. By using a mixed method design with interviews as the primary data source, a health measurement can be performed on data scarce ecosystems. This is applied to ResearchKit, Apple's SDK to create applications for medical research. The case study shows that the ResearchKit ecosystem is threatened by the outbound links of the third-party software developers. These developers intend to create web-based applications as ResearchKit suffers from a selection bias that makes it unsuitable for most medical research. The interviews exposed an inherent problem that is unrelated to ecosystem size and may not have been found in a traditional health measurement.

Keywords: Software ecosystems · Ecosystem health · Open source

1 Introduction

The rising collaboration of software companies with third-party developers allows software ecosystems to form. A software ecosystem is defined as "a set of actors functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts" [1].

Software vendors that have created a software ecosystem around their product have to rely on the software ecosystem to be successful. In order to ensure the success of a product, the ecosystem needs to be healthy, as ineffective use of a software ecosystem will lead to the demise of software vendors, as stated by Jansen et al. [2]. Therefore, to evaluate future success of an ecosystem orchestrator, a health measurement of the software ecosystem is essential.

The Open Source Ecosystem Health Operationalization (OSEHO) is a framework that provides a health measurement based on a list of metrics that differ for each software ecosystem [1]. However, to perform a health measurement based

A. Ojala et al. (Eds.): ICSOB 2017, LNBIP 304, pp. 131–145, 2017.
https://doi.org/10.1007/978-3-319-69191-6_9

on OSEHO, there currently is a focus on quantitative research methods. The quality of this analysis is highly dependent on the data and several challenges concerning unavailable, missing or incorrect data are reported [1].

Therefore, this paper proposes a new method of ecosystem health measurement. By interviewing developers, more in-depth information is to be found, which may predict the rise of an ecosystem. This way, not only well-established ecosystems can be assessed, but young and small ecosystems that are still data-scarce can be analyzed. In other domains, a mixed methods design has proven to provide both a "richer, contextual basis for interpreting and validating results" and "an increase of the robustness of results because findings can be strengthened through triangulation" as stated by Kaplan and Duchon [3]. This leads to the following research question:

> ***Research question:*** *H*ow can qualitative research methods be used to measure the health of open source software ecosystems?

This paper is written in the following structure. Section 2 reviews previous work and describes the framework for measuring ecosystem health. Section 3 describes the case study, research methodology and metrics. Section 4 covers the results of this research. In Sect. 5, the results are analyzed. Finally, Sect. 6 provides the discussion of the research, and in Sect. 7 the conclusion about software ecosystem health is made.

2 Previous Work

Software ecosystem health is a new research domain within software ecosystems. Few researchers discuss the specific topic within their work. A compact definition of software ecosystem health is given by Lucassen et al. [4] as "longevity and a propensity for growth." Some theories link software ecosystem health to biological ecosystem health, such as Dhungana et al. [5] and Wynn [6]. Newer research mainly focuses on extending the models of den Hartig [7] and Iansiti and Levien [8]. Den Hartig and Iansiti & Levien cover business ecosystems and recognize three determinants of ecosystem health. These are productivity, robustness and niche creation. The determinants are defined by den Hartig [7] as:

- robustness, the capability of an ecosystem to face and survive disruptions
- productivity, the efficiency with which an ecosystem converts inputs into outputs
- niche creation, the capacity to create meaningful diversity and thereby novel capabilities

Jansen [1] extends the model of Den Hartig by providing indicators for each of the determinants of open source software ecosystem health. Furthermore, the model is extended by adding two scopes, network level and project level. At network level, the determinants of ecosystem health are operationalized for the ecosystem domain. The project level covers determinants that investigate ecosystem health by analyzing projects within the software ecosystem. OSEHO has been used to measure the health of other software ecosystems such as e-commerce ecosystems by Alami et al. [9] and content management systems by van Lingen et al. [10]. The indicators are diverse as the availability of data for open source ecosystems is limited and different for each ecosystem [1]. Therefore, a refit of OSEHO to the analyzed framework is required.

3 Research Method

3.1 Case Study

The method for measuring ecosystem health of young software ecosystems is exemplified in a case study of the ResearchKit software ecosystem. ResearchKit[1] is an open source framework introduced and developed by Apple[2] that allows medical researchers and app developers to create applications for medical research. The goal of ResearchKit is to revolutionize the medical sector by having a software ecosystem that entails applications that give new medical insights on a faster scale than traditional medical research. The cooperation of Apple, third-party developers and medical researchers leads to a product that serves the market. Therefore, the environment of ResearchKit is defined as a software ecosystem. The open source code of ResearchKit has been released on GitHub in March 2015. However, the amount of data in this ecosystem is not sufficient to perform a traditional ecosystem health measurement. Therefore, the ResearchKit software ecosystem health should be measured by combining interviews and quantitative methods.

3.2 Ecosystem Health Metrics

The metrics used to measure the ecosystem health determinants in the interviews and GHTorrent search are discussed in this section. Because ecosystems and possible metrics differ, the metrics that should be used to measure the ResearchKit have to fit the ecosystem. The selected metrics are based on the 41 metrics of ecosystem health distinguished in the OSEHO framework [1]. For each metric included, the reason why it was added and a definition are shown below. Other metrics of OSEHO are not included due to a lack of data (such as new downloads), or because they are not practicable in this context (such as market share). The overview of the metric selection is shown in Table 1.

Some of the metrics are selected because of a pilot interview at C tracker. C tracker is an application within the ResearchKit ecosystem that is being used to gather medical data from Hepatitis C patients through the use of their smartphone. The pilot interview aimed to shed light on metrics of ecosystem health that are related to the third-party developers. Therefore, the metric **knowledge creation (1)** was added. Third-party developers should experience additional

[1] http://researchkit.org/.

[2] http://www.apple.com/researchkit/.

benefits of participating in the ResearchKit ecosystem in the form of easier subject collection, faster research paper development etc. This is an indication of the productivity of the ResearchKit ecosystem.

Usage (2) is defined as the number of end users of the released applications. Usage is added as a metric because the ecosystem can only exist when sufficient research subjects are available. The data about number of end users is an indicator of productivity and is only accessible through interviews, as Apple does not provide numbers of app downloads in its App Store.

The **growth of the software framework (3)** is measured by Van Lingen, Palomba and Lucassen [10] as the growth of the framework in modules. In the ResearchKit software ecosystem, this is measured as the number of commits to the software framework. The software framework is available on GitHub where Apple or other companies can edit and extend ResearchKit.

The total number of **active projects (4)** is measured based on the activity in the App Store. The total number of applications in the App Store is available, and it gives an indicator of the robustness of the ecosystem [1]. The number of active projects is a metric of ecosystem health, as it is a direct indicator of the size of the ecosystem. Lucassen et al., van Lingen et al., and Goeminne & Mens use this metric for ecosystem health measurement [4,10,11].

Contributor satisfaction (5) is the satisfaction of the developers of applications in the ecosystem. Developer satisfaction is supposed to be one of the most important metrics in project health as concluded by Lakhani and Wolf [12]. A high developer satisfaction binds developers to the ecosystem. Therefore, contributor satisfaction is an indicator of robustness.

The **end user rating (6)** is collected by scraping the App Store for ratings of end users. A high rating from end users is essential because they have to use the developed applications and participate in research to allow this ecosystem to function. Therefore, the end user rating is an indicator of the robustness of the ecosystem. Stoyanov et al. concluded that ratings can be used to measure the quality of mobile health applications [13], which is important for ecosystem health in this domain.

Outbound links to other SECOs (7) is defined as the other ecosystems where the contributors are active in. The pilot interview showed that the development team of C Tracker was also active with ResearchStack, the Android counterpart of ResearchKit. The multi-homing activities of developers may or may not be beneficial for the robustness of the ecosystem.

Interest (8) is measured both on developer and end user level. Search statistics using Google Trends are analyzed to measure the worldwide public interest in ResearchKit. This is done similarly to van Lingen et al. [10], who define the findability of Google Trends as an indicator of software ecosystem health when comparing several ecosystems. In this paper, the findability will be analyzed over time to measure the robustness in terms of public interest. Furthermore, developer interest is measured using the growth of the number of forks on GitHub. A fork is a separate repository where a third-party developer has full writing permissions. Forks may act as a first step to new projects [1]. Lucassen et al. [4] show that the number of forks is an indication of software ecosystem health.

Variety in projects (9) & **variety in developer type (10)** measure the niche creation of the ResearchKit ecosystem. Variety in projects measures what the goals and features are of the applications in this ecosystem. Variety in developer type discusses who the contributors to this ecosystem are. Their size and location may also influence ecosystem health. Iansiti & Levien state that a healthy ecosystem possesses the capabilities to increase meaningful diversity over time through the creation of new valuable functions [8]. In this article, it is argued that both variety in projects and developer type lead to an increased capability to create meaningful diversity in the software ecosystem.

Table 1. Overview of the selected software ecosystem health metrics

Determinant	Metric (number)	Source
Productivity	Knowledge creation (1)	Interviews
	Usage (2)	Interviews
	Growth of the software framework (3)	GitHub
Robustness	Active projects (4)	App Store
	Contributor satisfaction (5)	Interviews
	End user rating (6)	App Store
	Outbound links to other SECOs (7)	Interviews
	Interest (8)	Google Trends, Github
Niche creation	Variety in projects (9)	Interviews, Online search, App Store
	Variety in developer type (10)	Interviews, Online search, App Store

3.3 Data Collection

GHTorrent is used to obtain the historical evolution of the ResearchKit open source project. The GHTorrent project [14] provides queryable data offered through the GitHub REST API, created by the Software Engineering Research Group of TU Delft. The MySQL database is queried using the DBLite web-based client. For every GitHub commit to the ResearchKit project, the commit date, committer username and committer employer are retrieved. Next, all projects that were forked from the ResearchKit projects have been retrieved, together with their creation date, username and the user's employer. The data used in this work is dated 28 September 2016.

The **interviews** were conducted with key developers of ResearchKit applications. To find the developers of ResearchKit applications, the App Store was searched for these applications. In the App Store, 15 applications that used ResearchKit were found. The developers of these applications were sent an email request for an interview. A reply was received from eight developers (response rate 8/15 = 53%). Two of these developers did not have a final interview, as one of the developer teams replied that their application was used so rarely that they did not have enough information to give an interview and the other team did not reply to email response after the initial contact. Out of the remaining six developers, three developers were interviewed using Skype or Join.me. The other three developers answered the questions by email. The interviews had a semi-structured design and the interview questions were based on the metrics.

4 Results

In performing a qualitative ecosystem health measurement, the selection of ecosystem health metrics is the first step. An indicator should be selected when literature about the research topic states that the indicator is relevant for ecosystem health measurement. Then, data availability for each indicator has to be reviewed. Indicators that are not sufficiently covered by quantitative data sources are then selected to be measured in interviews. The interviews should be held at third party developers to measure the selected ecosystem health metrics. This can overcome the aforementioned problem of data scarcity when performing a software ecosystem health measurement. An operationalization of the measurement is shown below for ResearchKit.

4.1 Productivity

The interviews made clear that **knowledge creation** by developing an application with ResearchKit is significantly better than previous methods, such as paper questionnaires. The first advantage mentioned is that an application can provide validation steps and therefore reduce the occurrence of invalid data. The ResearchKit framework also allows for easy collection of sensory data. This is possible by manual coding but is easier when using ResearchKit. The lead software developer of C Tracker states:

> *You need to show surveys nicely on the screen and alternative solutions have not been nicely done; but ResearchKit is great. (...) So ResearchKit gives access to sensory data, you can do this yourself, but since ResearchKit provides it, you can more easily build an app around it.* - Lead Software Developer of C Tracker

Medical research using an application developed with ResearchKit is more effective than conventional methods, but also more effective than developing an application without ResearchKit. Furthermore, the research team that has developed the Mole Mapper application stated that "the first publication in a major journal was accepted," with data acquired through the Mole Mapper application.

Knowledge is not only created by the implementation of the ResearchKit framework itself, contributors are also adding knowledge to the ecosystem framework. This can be done by adding code extensions to the open source GitHub project, or sharing developer experiences, as explained by two interviewees:

> *We have sent pull requests that they accepted. It seems to work. (...) ResearchKit has a lot of contributors who have added back the active tasks modules.* - Lead Software Developer of C Tracker

> *We did contribute some of the forms to the community and did relate to Apple what we were doing and how we approach challenges that we faced. Other community members who were facing similar experiences now have a guide on how we solved that problem.* - Chief Information Officer of StopCOPD

The **usage** of the medical research applications varies. C Tracker mentioned 700 end users and Mole Mapper mentioned 3000 subjects in their first study. Bigger studies in terms of participants are mPower and PRIDE Study. mPower has reported more than 10,000 participants and PRIDE Study over 16,000. Another developer reveals that their application has not been used very much, as the developer was unable to get the application out to patients to try. The development team of StopCOPD already had a web-based platform and the release of an application developed with ResearchKit did not significantly impact usage.

Figure 1 shows the **growth of the software framework** in the green area that is expanding over time. Figure 2 shows the contribution of Apple to ResearchKit in comparison with other software companies.

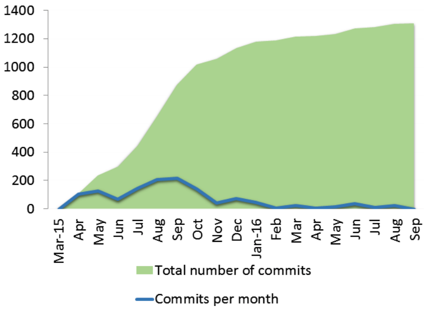

Fig. 1. Number of commits per month and cumulative number of commits to ResearchKit (Color figure online)

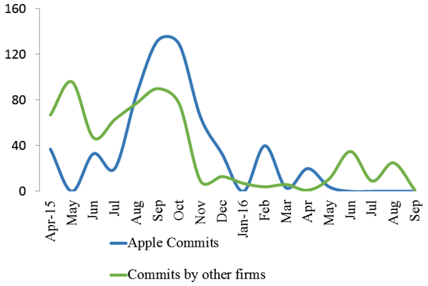

Fig. 2. Commit comparison of Apple and other firms over time

4.2 Robustness

Out of the 15 **active projects** listed in the U.S. App Store's ResearchKit page, four apps have been updated in the last six months. Out of the other applications, nine have been updated in the last year and two apps have not been updated for more than a year. In addition to this, two applications that have been launched in the first wave of ResearchKit apps have been retracted from the App Store.

The **contributor satisfaction** of interviewed developers with the use of ResearchKit was overall high. Using this framework speeds up the development process and ensures consistency throughout the ecosystem.

> *So you have a nice app and a really nice package that looks good and works well. But it's not something that you are unable to do without ResearchKit. It's just nicer and more quickly done.* - Lead Software Developer of C Tracker

Another aspect of ResearchKit that positively impacted contributor satisfaction is the fast collection of research subjects. The development and publication of a mobile application proves to be a fast way of collecting subjects and consenting them to participate in research. The principal investigator of PRIDE Study was satisfied with the faster development of a medical research application and the digital consent.

> *It allowed us to recruit people quickly by having an app. People have to consent to participate in research. That is done through the app which is relatively new in this whole process.* - Co-Director and Principal Investigator of PRIDE Study

ResearchKit does have aspects that negatively impact contributor satisfaction. Bugs were found by developers that would not occur when an application is developed without ResearchKit.

> *We ran into this issue with ResearchKit which is a bug in the branching logic of surveys. The developers had to do coding of the correct order both forwards and backwards through the survey, which I think was just really annoying. As ResearchKit develops more, it will be hopefully less of a problem.* - Co-Director and Principal Investigator of PRIDE Study

The **end user rating** is measured by App Store ratings. These are shown in Table 2. The score is measured on a scale from one to five. In order to measure an average rating of the developed applications, only applications with five or more ratings have been taken into account, to reduce the influence of individual ratings on infrequently rated applications. These applications received 60.67 (SD = 71.02) ratings on average. The users scored the applications with an average score of 3.21 (SD = 0.59).

The **outbound links to other SECOs** are present on a large scale in this ecosystem as all interviewees reported that their research team did some kind of multi-homing. The goal of the medical studies is to reach as many subjects as possible, and developing applications for multiple platforms is a way to reach more subjects. The Android counterpart of ResearchKit is ResearchStack[3], an upcoming development that also provides a framework for developing medical research applications. ResearchStack is used by several of the interviewed developers.

> *Our developers are also working with ResearchStack. We have worked with this system to port one study over to ResearchStack and are in the process of doing so for additional studies.* - Principal Investigator (anonymized)

[3] http://researchstack.org/.

Table 2. Applications using the ResearchKit framework, listed on the ResearchKit overview page in the U.S. App Store

App name	Developed by	Launch	Last updated	Number of ratings	Score *****
MyHeart Counts	Stanford Medicine	Mar 2015	Dec 14 2016	225	2.91
PRIDE Study	University of California	Jun 2015	Jul 25 2016	99	2.86
GlucoSuccess	Massachusetts General Hospital	Mar 2015	May 13 2015	93	2.78
SleepHealth	American Sleep Apnea Association	Mar 2016	Jun 24 2016	46	2.78
Parkinson mPower Study	Sage Bionetworks	Mar 2015	Mar 21 2016	43	3.60
Mole Mapper Melanoma Study	Sage Bionetworks & Oregon Health & Science University Dermatology	Oct 2015	June 4 2016	13	3.62
EpiWatch	Johns Hopkins University	Oct 2015	Feb 19 2016	12	2.42
PPD ACT	Psychiatry and Genetics at the University of North Carolina	Mar 2016	Jun 27 2016	9	4.22
Autism & Beyond	Duke University Health System	Oct 2015	Nov 11 2015	6	3.67
VascTrac	Stanford Cardiovascular Institute	Sep 2016	Sep 22 2016	2	5.00
TeamStudy	Harvard & Sage Bionetworks	Mar 2016	Jan 4 2017	2	5.00
Concussion Tracker	NYU Langone Medical Center	Nov 2015	Feb 1 2016	1	5.00
C Tracker	Boston Children's Hospital	Oct 2015	Feb 15 2016	0	–
EPV	Yale University	Oct 2015	Jan 29 2016	0	–
StopCOPD	COPD Foundation	Nov 2015	Jun 29 2016	0	–

> *We find it critical to allow each ResearchKit study to be ported to Android phones using tools such as ResearchStack. The heterogeneity of Android devices makes Android much more challenging. We are outlining how to approach ResearchStack, and it is definitely in the works.* - Principal Investigator of VascTrac

Another way to target more subjects is by eliminating the use of a smartphone by developing a web-based service. This service is accessible through multiple devices ranging from computers to phones. This also eliminates a selection bias. When using an application developed with ResearchKit for medical research, the subjects are automatically users of iPhones, and this will bias the sample. Using a web-service with multi-device access eliminates this bias.

> *In the United States iPhones are owned, by effectively well-to-do people and we don't have as much racial and ethnic diversity as we would like. (...) So we're moving to a web-based platform that will allow people on any device to participate.* Co-Director and Principal Investigator of PRIDE Study

Figure 3 shows the number of forks created by developers on GitHub. The release of ResearchKit sparked the **interest** of developers and 484 forks were made in the first three months. After the initial three months, the number of created forks per month declined to an average of 10 forks per month.

The (end user) interest in ResearchKit, based on data from Google Trends[4], is visualized in Fig. 4. The numbers show that the search terms ‘ResearchKit’ and ‘Research Kit’ have a 2 till 3% popularity in the last quarter of 2016, in comparison with its peak popularity during the launch in March 2015.

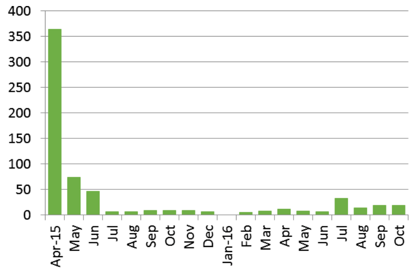

Fig. 3. Number of forks per month on the ResearchKit GitHub project

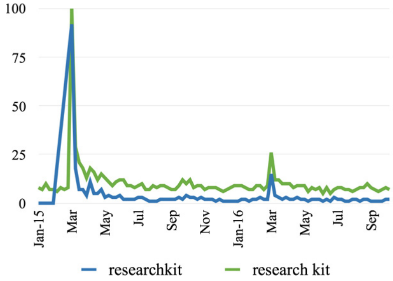

Fig. 4. Google search interest in ResearchKit

4.3 Niche Creation

The **variety in projects** developed with ResearchKit is high. Out of the apps mentioned in Fig. 2, 12 are developed for medical research dedicated to a disease. The other applications conduct research on related medical subjects, not specifically dedicated to a single disease. The interviews show different kinds of research conducted with ResearchKit. The first type of research aims to find a relation between people’s perceptions or way of living and a disease. For example, C Tracker, GlucoSuccess and MyHeart Counts are applications that try to analyze a subject’s behavior and link this to their disease. The second type of research tries to build up a community of like-minded and learn from their needs, of which PRIDE study is an example. These different types of research demand for as many subjects as possible, but also require an obtained sample to be representative of the population intended to be analyzed. A third study design is more exploratory in the sense that it tries to determine whether the iPhone’s sensory data could be helpful as a research instrument. One example is the EpiWatch app, which gathers data to facilitate the creation of a seizure detection app. The Parkinson mPower Study is also using this study design. When developing an application with an exploratory goal, a large data set is still vital, but a non-random sample is not a requirement for this research.

Table 2 also shows the developers of the applications in the App Store, in order to analyze the **variety in developer type**. The developers are not new start-ups that wish to enter the medical research sector. Instead, the developers

[4] https://www.google.com/trends/.

consist of universities, hospitals, foundations and nonprofit organizations. The organizations only take part in the ecosystem with a single application, except for Sage Bionetworks, a nonprofit organization that has developed three applications.

5 Analysis

The metrics that measure **productivity** address a variable change over time [1]. In the software ecosystem of ResearchKit the growth of the software framework is such a variable. The core framework was provided and enhanced in its first year primarily by Apple, but additions afterwards are mostly added by third-party developers. Commit access to the ResearchKit project repository is strictly controlled by the use of pull requests, consistent with findings from Padhye et al. [15]. Contributors have to sign a license agreement first and do not have direct writing access to the source code. The initial framework appears to be finished, as Apple is no longer committing intensively and the number of commits per month over the last year is significantly lower than in earlier stages of development.

The great variation in the number of end users between the applications emphasizes the notion that ResearchKit as a technical framework alone provides little advantage over an application created without the framework, in terms of recruitment. Still, some applications profited from the media attention during the launch of ResearchKit and mention over 16,000 downloads and another development team thinks it is easier to recruit people by having an application. ResearchKit may have been an incentive for them to create such an application.

Knowledge creation can substantially be increased by developing an application with ResearchKit, due to the predesigned surveys, sensory data and digital informed consent. This is true when comparing to traditional medical research methods and when comparing to development of a research application without ResearchKit.

The **robustness** level covers absolute entities in static metrics [1]. A robust ecosystem is able to “survive disruptions and must persist in the face of environmental changes” [8]. Currently, the number of active ResearchKit projects is limited to 15 applications highlighted by Apple in their U.S. App Store. The exact number of ResearchKit applications worldwide is higher and not all applications are officially recognized in the App Store ResearchKit category. Nevertheless, both end user and developer interest declined significantly after the first three months following the launch of the framework. Over half of the created related projects in ResearchKit’s 1.5 years appearance on GitHub are forked within the first month after release.

Paschou et al. [16] emphasize an increasing demand for “continuous software updates of mHealth apps on users’ smartphones.” Developers mostly meet this requirement, as a new version has been released in the last year for 13 out of the 15 applications. End users rate the applications with an average 3.21 score on a scale from one to five, but the number of ratings applications have received varies substantially. More important for the robustness of the ecosystem is the developer satisfaction and their connection to the ecosystem. Iansiti & Levien [8] even state that niche players (the developers) can wield surprising power in the face of keystones (Apple). In the ResearchKit ecosystem, developers are overall satisfied with the technical functioning of the framework. Minor bugs in the framework or irritations of developers have been attributed to the ecosystem still being nascent.

However, serious threats to the health of the ecosystem concerning its robustness have been identified. These threats come from niche players' outbound links to other ecosystems. Almost all of the interviewees raised concerns about the arising sample selection bias due to the fact that ResearchKit applications can be used by iPhone users only. Moreover, medical researchers tend to look for alternative solutions to overcome this problem, clearly harming the ResearchKit ecosystem. ResearchStack, the Android counterpart of the framework, is proposed and used by some researchers as a method to reach more possible subjects. This open source framework is especially designed to easily adapt existing iOS apps for Android.

Another strategy involves the development of a web application, resulting in the iPhone app becoming only little more than a 'wrapper' for the mobile website. These changes can "loosen the bonds that typically tie a niche player to its keystone partner and make it easier for developers to end a relationship with a keystone whose platform doesn't offer sufficient value" [8].

Next to the productivity and robustness, **niche creation**, an ecosystem's capability to "increase meaningful diversity through the creation of valuable new functions or niches" is an important health measurement [8]. In the ResearchKit ecosystem the projects have widely varied goals. There is no overlap among the projects and currently no competition. This is why new players can easily become active in a new domain or niche. The app initiators come from several sources, including universities, hospitals and non-profit organizations. New entrants already have a relation to medical research, indicating that this market can be characterized as one with high entry barriers. Despite these entry barriers, there is much opportunity to start as a new niche player in the ecosystem because of the large variety in projects on the network level.

On the project level, Jansen [1] makes clear that "a project that can be applied in a wide variety of contexts will be more supporting for niche creation." This comprises that the project should be able to support different languages, markets and technologies. In ResearchKit, these possibilities are currently limited. The technology and market are limited to the use of the iOS developer platform and additional countries are only slowly being adopted due to the strict regulations on medical research that vary for each country. From the results of this paper, three different ResearchKit practices became clear. Traditional medical research, aiming to find a relation between people's perceptions or way of living and a disease, is supplemented with community building applications and exploratory research to the usefulness of iPhone's sensory data for future implementations.

Based on the productivity, robustness and niche creation metrics, this analysis shows that the ecosystem is unhealthy. Although the low scores, mainly on robustness, seems to be attributable to the small size of the ecosystem, the interviews show clear indications of an underlying problem that is unrelated to ecosystem size.

The main finding of this analysis is that all of the third-party developers in the ResearchKit ecosystem have outbound links to other SECOs. Especially developers that have created or are intending to create a web-based application are threatening ecosystem health. These developers are leaving the ecosystem because of the sample bias that persists when developing medical research applications for the iPhone. ResearchKit is unsuitable for medical research in which normal sampling is required. A small number of research applications in the ecosystem is related to exploratory research to test the usefulness of the iPhone's sensory data for medical purposes, which is not stymied by sampling bias. However, this exploratory research does not appear to be large enough to create an ecosystem around these applications. The main advantage that ResearchKit has over a web-based application, the access to sensory data, has not been widely adopted by the actors in the ecosystem.

Iansiti and Levien [8], who define value creation as the first part of an effective keystone strategy, emphasize that "value creation of keystones is crucial to the community's survival." The need for standardized technologies (like HTML5), follows logically after the initial industry-specific technology and tailored software solutions [17]. This is underlined through the need for random sampling in medical research. Since the added value of sensory data is not high enough when compared to the need to retrieve a non-biased sample, the ResearchKit framework in its current shape will be unable to establish a healthy software ecosystem around itself.

6 Discussion

This article aimed to create a health measurement that is not dependent on large quantitative data sources by introducing interviews as a new data source. This new technique has been applied to the ResearchKit software ecosystem. In this chapter, the limitations and implications of this research are discussed.

Using both qualitative and quantitative data sources for ecosystem health measurement provides the best of both worlds. Interviews give in-depth data and insights that may not be easily retrievable from quantitative sources. In the ResearchKit case study, the outbound links of third-party developers may not be retrieved from quantitative data sources. Quantitative methods provide objective data that is easier to collect and larger in volume. Furthermore, quantitative data sources can provide information of the ecosystem over time. Qualitative and quantitative sources complement each other in the analysis of software ecosystems. Trends in interviews may be linked to findings in quantitative methods and vice versa, which strengthens the validity of the analysis.

The method of this paper is especially useful for ecosystems where no plethora of data is available. This can be the case for newly started ecosystems, such as ResearchKit. The method may also find use in closed software ecosystems that have just been opened up. The list of metrics is created with the definition of the health criteria as defined by Jansen [1] in mind. The interviews can theoretically provide information on limitless metrics. To ensure the quality, only few predefined metrics should be researched based on the specific case.

7 Conclusion

Open source software ecosystems are a new research domain within information systems. This paper attempts to contribute to the fresh domain by proposing a new method of measuring ecosystem health of data-scarce ecosystems. The research question of this paper is: How can qualitative research methods be used to measure the health of open source software ecosystems?

To answer the research question, a method that combines qualitative and quantitative data collections for an ecosystem health measurement has been proposed. Based on the OSEHO framework and relevant literature, ecosystem health metrics should be selected that measure the three determinants of ecosystem health: productivity, robustness and niche creation.

The method is operationalized in a case study on ResearchKit. The ecosystem health is measured by combining data retrieved from GHTorrent and interviews at developers of applications in the ecosystem of ResearchKit. The case study shows how interviews provide another perspective of ecosystem health than quantitative methods. The robustness of the ResearchKit ecosystem was found to be severely limiting its health, regardless of the size of the ecosystem. This is caused by the outbound links of third-party developers. Using ResearchKit in development for medical research applications implies that the subjects are iPhone users. This is restraining medical research that relies on unbiased random selection. Therefore, third-party developers have created, or are intending to create, a web-based application or adapt their applications for Android to reach a bigger audience with a smaller sampling bias. The interviews shed light on aspects that may not have been found in a traditional health measurement.

Further research could investigate the impact of ecosystem health on commercial success, because a link between these terms will lead to better understanding of ecosystem health. Another research area is to compare ecosystem health to external factors, as current ecosystems that appear healthy may still be quickly overtaken by superior ecosystems.

Acknowledgements. We would like to thank Prof. Dr. João Fernandes and Dr. Andrey Saltan for providing valuable comments that have improved our paper.

References

1. Jansen, S.: Measuring the Health of Open Source Software Ecosystems: Moving Beyond the Scope of Project Health (2013)
2. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: a research agenda for software ecosystems. In: 31st International Conference on Software Engineering, New and Emerging Research Track, pp. 187–190 (2009)

3. Kaplan, B., Duchon, D.: Combining qualitative and quantitative methods in information systems research: a case study. MIS Q. **12**, 571–586 (1988)
4. Lucassen, G., van Rooij, K., Jansen, S.: Ecosystem health of cloud PaaS providers. In: Herzwurm, G., Margaria, T. (eds.) ICSOB 2013. LNBIP, vol. 150, pp. 183–194. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39336-5_18
5. Dhungana, D., Groher, I., Schludermann, E., Biffl, S.: Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, pp. 96–102. ACM (2010)
6. Wynn, D.: Assessing the health of an open source ecosystem. In: Emerging Free and Open Source Software Practices. Idea Group Publishing (2007)
7. den Hartigh, E., Tol, M., Visscher, W.: The health measurement of a business ecosystem. In: Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting, pp. 1–39 (2006)
8. Iansiti, M., Levien, R.: Strategy as ecology. Harv. Bus. Rev. **82**(3), 68–78 (2004)
9. Alami, D., Rodríguez, M., Jansen, S.: Relating health to platform success: exploring three e-commerce ecosystems. In: Proceedings of the 2015 European Conference on Software Architecture Workshops, pp. 43–49. ACM (2015)
10. van Lingen, S., Palomba, A., Lucassen, G.: On the software ecosystem health of open source content management systems. In: 5th International Workshop on Software Ecosystems (IWSECO 2013), pp. 38–51 (2013)
11. Goeminne, M., Mens, T.: A comparison of identity merge algorithms for software repositories. Sci. Comput. Program. **78**(8), 971–986 (2013)
12. Lakhani, K., Wolf, R.: Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. MIT Press, Cambridge (2005)
13. Stoyanov, S.R., Hides, L., Kavanagh, D.J., Zelenko, O., Tjondronegoro, D., Mani, M.: Mobile app rating scale: a new tool for assessing the quality of health mobile apps. JMIR mHealth uHealth **3**(1), e27 (2015)
14. Gousios, G.: The GHTorrent dataset and tool suite. In: MSR 2013, pp. 233–236 (2013)
15. Padhye, R., Mani, S., Sinha, V.S.: A study of external community contribution to open-source projects on GitHub. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 332–335. ACM (2014)
16. Paschou, M., Sakkopoulos, E., Tsakalidis, A.: easyHealthApps: e-Health apps dynamic generation for smartphones & tablets. J. Med. Syst. **37**(3), 9951 (2013)
17. Tyrväinen, P., Warsta, J., Seppänen, V.: Evolution of secondary software businesses: understanding industry dynamics. In: León, G., Bernardos, A.M., Casar, J.R., Kautz, K., De Gross, J.I. (eds.) TDIT 2008. ITIFIP, vol. 287, pp. 381–401. Springer, Boston (2008). doi:10.1007/978-0-387-87503-3_22