Health Vulnerabilities in Software Ecosystems: Five Cases of Dying Platforms

Master Thesis

Lamia Soussi
5582253
l.soussi@students.uu.nl

Supervisors:
Dr. R.L. (Slinger) Jansen
slinger.jansen@uu.nl

Dr. S. (Sergio) Espana Cubillo
s.espana@uu.nl

Universiteit Utrecht

Master of Science in Business Informatics
Department of Information and Computing Science
Faculty of Science

Abstract

Software ecosystems are networks of organizations working together to collaboratively serve a market. A software ecosystem is considered healthy when it is productive, robust and is growing on increasing numbers of partnerships, or acquiring new members. One can also consider an ecosystem unhealthy, when it is not satisfying its customers. In this article, we examine five cases of unhealthy or dying ecosystems through a qualitative analysis of its community forums and its presence on social media. We extract five indicators for an ecosystem's demise and their countermeasures for avoiding an unhealthy ecosystem. With these indicators and countermeasures, we can help practitioners avoid pitfalls that are faced by their business and overall ecosystem, and finally help them make strategic decisions.

Keywords: Software Ecosystem Health, Platform Ecosystem, Unhealthy Ecosystem, Big data analytics, Ecosystems analytics

Acknowledgements

This Master's thesis is the result of a research project that lasted for several months. Throughout this period, I have been accompanied and supported by numerous people and I am thankful for their support. First of all, I would like to thank my first supervisor Slinger Jansen. Slinger, thank you for your intensive feedback sessions through Skype, phone and face-to-face meetings, and I am grateful for your contagious enthusiasm and your patience in the time-consuming execution of this project. Next, I would like to thank my second supervisor Sergio Espana Cubillo. Sergio, thank you for the feedback, reviews and advice. I am grateful for your guidance, which led to creating a constructive and well guided report. I would also like to thank the other staff members from the Organization and Information research group of the Master Business Informatics, I really appreciated the study program, the input during lectures and the contagious passion everyone expressed in their respective fields. Finally, my gratitude goes to my partner, friends and family. They have surrounded me for many years, supported and encouraged me whenever necessary. I am very grateful for all of that.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"A set of businesses 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."

This definition of software ecosystems (SECO) is introduced by Jansen, Finkelstein and Brinkkemper in (2009). Since then, research about the subject extended, and to define software ecosystems multiple researchers have worked on it ((Lungu, 2009); (Bosch & Bosch-Sijtsema, 2010); (Manikas & Hansen, 2013b)), each providing their own definition aligning with their own goals.

Soon enough the notion became more common among stakeholders, who started studying the ecosystem they belong to in order to control their market view and business expansion (Manikas & Hansen, 2013a). However, the status of an ecosystem can differ over time. At the start of every big activity; for instance a new project or a new business venture, the future of a company and therefore its whole ecosystem is unpredictable. To this extend, an ecosystem can be studied in measure of health which is of major importance when stakeholders want to find out their position in the ecosystem and in the market.

The notion of ecosystem health initially comes from natural ecosystems. Costanza (1992) refers to a healthy ecosystem as "being 'stable and sustainable'; maintaining its organization and autonomy over time and its resilience to stress". Additionally, Rapport, Costanza and McMichael (1998) introduced three health indicators of natural ecosystems; vigor, organization and resilience. Vigor refers to 'activity, metabolism or primary productivity'; organization is presented as "the diversity and number of interactions between system components"; and resilience as the capacity to maintain structure and function in the presence of stress. Analogically, we find the health of an ecosystem in the information science domain defined by Iansiti and Levien (2004) introducing three different but somewhat similar terms: productivity, robustness, and niche creation. The fist one, productivity, refers to new projects created and the changes applied over time. Second, robustness refers to the status of established projects in an ecosystem that are still active and the survival of the ecosystem after disturbance. Finally, niche creation refers to the diversity provided by an ecosystem that could lead to innovations in technology.

In this paper, we theorize that failing ecosystems or parts of ecosystems (e.g. platforms, projects) follow a pattern that leads to their unhealthiness. Therefore, we are looking at the weak spots and threats that lead to an unhealthy ecosystem. Finding those weaknesses requires determining certain behaviors or links generated by actors or procedures in the life of an ecosystem.

To keep the analogy between natural ecosystems and software and business ecosystems Fa et al. (2014) define weak spots in their study of biodiversity as "high diversity regions of hunting vulnerability for wildlife". In our research which focuses on software ecosystem health, we refer to the 'so-called' weak spots in natural ecosystems as "health vulnerabilities". And therefore, propose the following definition:

"An ecosystem's health vulnerability is any circumstance that may expose an ecosystem to the possibility of being harmed and become unhealthy".

1.1 Problem statement

Project failure and critical success factors have been studied by managers in the past decades. Pinto and Mantel (1990) and Poon and Wagner (2001) determine causes and factors through cases in a managerial perspective following previous work of older studies. In our perspective, we relate those factors to the social and business aspect by looking at the ecosystem and the interaction between its actors. To the best of our knowledge, this aspect has not been investigated so far.

Because stakeholders are not necessarily aware of the paths that could lead to the failure of certain projects, we identify what we defined earlier "health vulnerabilities" and provide guidelines to avoid making the wrong business decisions before it becomes irreversible. To find out how an ecosystem becomes unhealthy, we require literature review and collection of data.

Most ecosystem research studies have focused on the impact on an ecosystem and it's health using data from the business side or on the developers side. Certain literature present their findings by studying application stores or repositories such as Github, analyzing for instance the number of downloads or spin-offs among others (Lucassen, van Rooij & Jansen, 2013; Hyrynsalmi et al., 2012; Hyrynsalmi, Suominen & Mäntymäki, 2016; Soussi, Spijkerman & Jansen, 2016; Kabbedijk & Jansen, 2011). Other studies, include systematic literature review (SLR) or a combination of SLR and expert interviews (Van Den Berk, Jansen & Luinenburg, 2010; West & Gallagher, 2006; Wnuk, Runeson, Lantz & Weijden, 2014).

However, some of them state the importance of customers in an ecosystem and furthermore its health; *"The success of an ecosystem is dependent on both developers and customers"* (Hyrynsalmi et al., 2016) , *"Customers or buyers are one of most important, if not the most important, forces on a SECO"*(Van Den Berk et al., 2010) , *"End customers are the largest group of ecosystem participants who indirectly influence the evolution of the ecosystem via their requirements and needs"* (Wnuk et al., 2014).

These statements raise our interest, since a great number of studies about ecosystem health acknowledges the importance of the customers, but to our knowledge no research has been conducted using their perspective. Therefore, we analyze the end user perspective by studying the user community and their communication among each other and towards the rest of the ecosystem.

1.2 Research questions

The goal of the current study is to provide stakeholders, mainly new and future business and project owners, with practical knowledge to avoid the pitfall of an unhealthy ecosystem before the launch of their project or company. This knowledge is derived from the customer base, which is in our opinion the reason why companies keep existing in the first place.

Throughout our paper, we are answering the following main research question (MQ):

- **MQ: How can software ecosystem health vulnerabilities be identified and eliminated?**

To answer our main research question we answer first those following sub-questions:

- **Q1: What are the characteristics of healthy and unhealthy ecosystems?**

  Software ecosystems and their health have been studied by researchers for several years. Frameworks have been established in different studies targeting different types of ecosystems (i.e. Business, software, open source). We aim at finding out the differences and most importantly the similarities between those frameworks and what can help us achieve our goals at finding vulnerabilities in ecosystems.

- **Q2: Which unhealthy ecosystems can be studied and how to choose them?**

  The health frameworks investigated through literature review while answering the sub-question Q1, allowed us to identify a set of health metrics that highlight which parts of an ecosystem tend to give away it's health status. Therefore, allowing us to determine the health of chosen cases, and precisely where to identify their vulnerabilities.

- **Q3: What are observable weaknesses in software ecosystems?**

  The answer to this question is provided through a step by step approach. First, data is collected from our case studies. Then, analyzed to identify repeated characteristics which highlights observable patterns. And finally the health vulnerabilities are derived in support with the literature and the case studies.

In order to answer the research questions, we are considering a qualitative approach by conducting literature review of related studies and theory building case studies including data analysis. First, literature is selected to construct the knowledge in the domain of health ecosystems and measuring it. Second, a number of case studies are selected based on health metrics identified from the first step. Those case studies revolve around failed projects that died soon after their launch or ecosystems that unexpectedly did not succeed. Last, we collect data from different portals related to each case study, and analyze it.

Table 1.1 presents which research method and analysis method are used to answer each research question, and what are the deliverable in each section.

Research Question	**Data collection method**	**Data analysis method**	**Resulting deliverable**
Q1	Literature review	Keywords-in-context, Qualitative comparative analysis	List of frameworks and metrics
Q2	Literature review, Data mining	Qualitative content analysis	List of unhealthy ecosystems
Q3	Data mining	Qualitative content analysis, Statistical analysis	Noticeable patterns
MQ	Empirical evidence	Cross-Case Analysis	Identified Health Vulnerabilities and Guidelines

Table 1.1: Research question and methods matrix

1.3 Relevance

1.3.1 Scientific Relevance

Since the research domain of software ecosystem is relatively recent, opportunities are left unstudied and provide potential in both research and industry (Barbosa & Alves, 2011; Manikas & Hansen, 2013b).

A large number of research studies have been conducted on open source ecosystems, modeling and ecosystem health (Franco-Bedoya, Ameller, Costal & Franch, 2017). A large proportion consists of case studies on a single ecosystem studied form the developer network perspective. A limited number of studies have been made using multiple cases at once. And no study involves detecting health threats rather than a health framework.

Our research will give an insight on what threatens an ecosystem's health using the perspective of one of the most important actor of a SECO, the customer (Van Den Berk et al., 2010). Furthermore, this research includes multiple case studies on different types of platforms, resulting in findings adaptable for multiple domains of ecosystems.

The results of our research could be useful for predicting the evolution of an ecosystems in their early stages and the continuity of it's health.

1.3.2 Practical Relevance

Apart from its scientific relevance, our research has several benefits for practitioners. Our study facilitates the identification of health vulnerabilities and means to eliminate them. Furthermore, this research gives an insight into the impact of an underrated actor (i.e. the end user) on the ecosystem and it's health. Knowing the vulnerabilities identified from our case study, practitioners have a clear view of what should be pursued and what should be avoided helping them make strategic decisions in the development of software and platform projects.

1.4 Document Structure

This introduction chapter is followed by defining our research method in chapter 2. Each part of the research method is then dissected in individual chapters starting with the literature review in chapter 3 which contains the theoretical background helping us answer our first sub question. Then, in chapter 4 we discuss the case study design answering our second sub question on how we chose our case studies and detect health vulnerabilities. The data collection is provided in chapter 5, detailing the steps taken to collect the data from each source. The analysis of the data and it's interpretation is presented throughout chapters 6 providing our deliverable and finally answering our main research question. We discuss our results and limitations in chapter 7 and finalize the document with a conclusion and suggestions for future research.

Chapter 2

Research Method

2.1 Literature Study

The first part of this research is a literature study performed by reviewing previously published scientific studies from sources such as scientific journals and conference papers. Since a state of the art study is required to form our goals and theory, we start analyzing our selected literature using keyword-in-context analysis in order to identify studies on certain subjects (e.g. failing ecosystems, software ecosystem health, data analysis, case study research) to understand its meaning across sources (Onwuegbuzie, Leech & Collins, 2012). A literature database on the study of the health of ecosystems is formed and analyzed using qualitative comparative analysis. This method systematically analyzes similarities and differences across publications, in order to assess causality in findings across sources (Onwuegbuzie et al., 2012).

2.2 Case study Design

A case study research can be descriptive, exploratory and explanatory (Yin, 2009). The purpose of our research is to identify health vulnerabilities and get insight on how to avoid or solve them.

We are using exploratory case studies with qualitative and quantitative analysis. Data is collected to look at phenomena of real life events determining patterns among different case studies that will highlight practices that threaten the health of software ecosystems.

Both quantitative and qualitative data are included in the case study methodology to help explain both the process and outcome through complete observation, reconstruction and analysis (Tellis, 1997).

With respect to the case study research process presented in figure 2.1, we started by mentioning our theory and research questions in the first chapter. The selection of the cases is further discussed in chapter 4. Data analysis and cross-case analysis is presented in chapter 6.

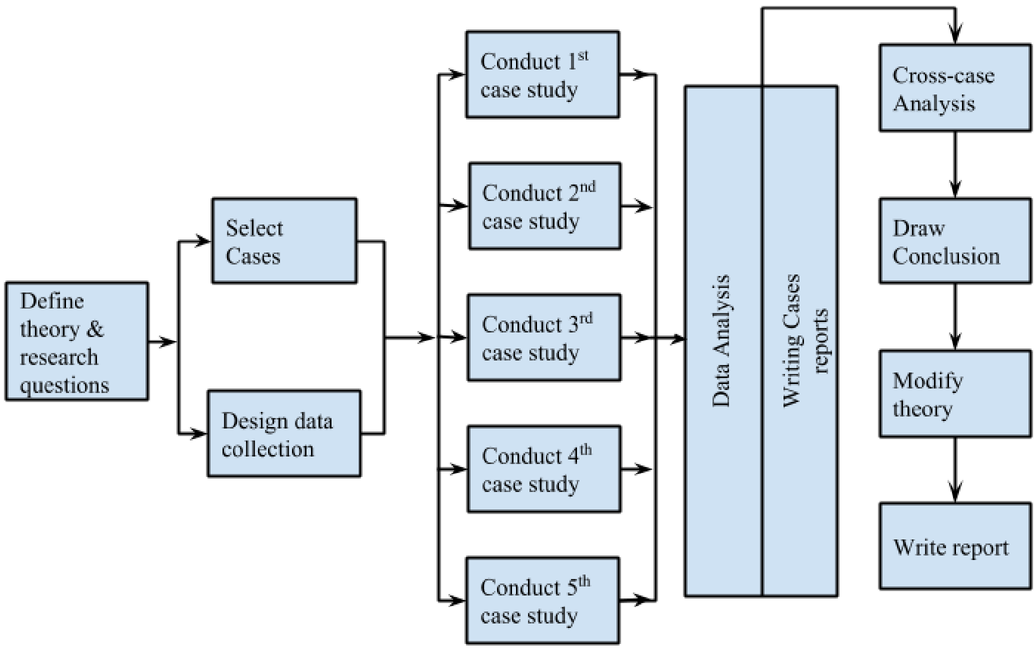


Figure 2.1: Case studies research process, adapted from Yin, 2009

2.3 Data Gathering

To increase validity of the research, we aim to triangulate our research techniques by using related literature, data mining, and data analysis, and focus our target on developers and customer experiences.

Measuring the relationships in software ecosystems is often established by collecting data from a domain-specific source, for instance, studying the developer community, including the way actors cooperate, communicate and share information. For our research goal, we decided to use the Jupyter notebook since it provides the ability to mine different website with one single tool.

The Jupyter Notebook was commonly used under the name of the IPython Notebook. With the use of a virtual machine and a vagrant file, a user can mine websites using Python code text. A web server is running on the virtual (guest) computer to allow the user to access the notebook via port forwarding from the host computer. In his book, Russell (2013) offers directives to mine the social web data through illustrative and concise code, by first explaining how to install and use the tool and then by building up example of codes. The second edition of the book includes GitHub in addition to Facebook, Twitter, LinkedIn, Google+ and web pages. This aspect of mining several types of websites with the same method is appealing since we can have the customer's view of the ecosystem, the business view and the developer's view.

In chapter 5, we explain which sources are most useful to answer our research question and to provide our findings. Furthermore, a detailed method of collection and its limitations is explained per source.

2.4 Data analysis

The gathered data is analyzed using qualitative content analysis. This method involves systematic reading or observation of texts or artifacts to indicate interesting and meaningful patterns (Neuendorf, 2016).

With the use of the tool NVivo[1] our data can be stored, organized, categorized, coded and analyzed. Nodes are created in order to determine the recurrent subject within all data documents. A node, to not confuse with ecosystem nodes, is a collection of references about a specific theme, case or relationship (Help-nv11.qsrinternational.com, 2018 (accessed April 4, 2018)).

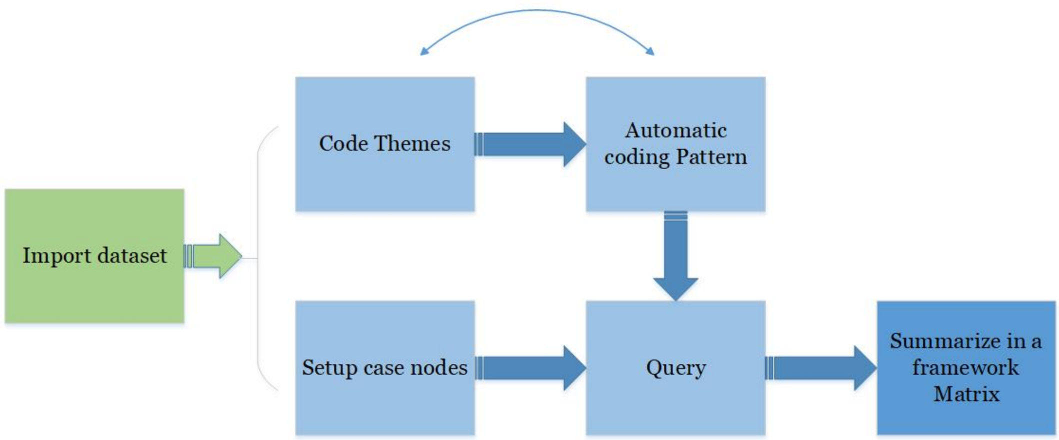


Figure 2.2: Nvivo Data Cycle

Figure 2.2 represents the data cycle on Nvivo from input to process and finally output. In the coding process, we analyze the subjects by selecting all phrases or paragraphs pointing at a certain issue. Those issues are categorized in multiple themes. Those themes are created taking into account the adaptation of health metrics extracted from the literature review.

Due to the extensive data, we make use of the automatic coding pattern using a predetermined coded part of the data. This feature of the NVivo tool is particularly useful for large data sets. The auto coding makes use of existing coding patterns, which represents initial coded text. We start by coding up to 5 pages from all the data documents, which approximately covers 1000 subjects per document.

The automatic coding using existing pattern follow the steps bellow in order to code the rest of the pages in our data set (Help-nv11.qsrinternational.com, 2018 (accessed April 11, 2018)).

1. Each node is broken down into a list of words. During this process, stop words are removed from the list. For stemming languages, words with the same stem—for example house, houses and housing—are grouped together.

2. For each term (group of words with the same stem) within the node, the 'term frequency-inverse document frequency' (tf-idf) is calculated.

3. In this context, the tf-idf[2] reflects how important a term is to a specific node. It is based on a term frequency calculation within the node that is offset by the frequency of the term in all of the nodes.

4. The list of terms and their associated tf-idf values are then represented as a one-dimensional array—known as a vector—for each node. The vectors for each node are used later on when determining what to code.

[1] http://www.qsrinternational.com/nvivo
[2] http://www.tfidf.com/

Chapter 3

Literature Study

In this chapter we discuss the literature review conducted to define ecosystem health and ecosystem death. Additionally, we provide a comparative study of existing health frameworks of different types of ecosystems. Furthermore, this literature study helps us determine the criteria to later choose our "dead" or "dying" ecosystem cases.

3.1 Defining Ecosystem Health and Ecosystem Death

As previously mentioned in our introduction, Costanza (1992) refers to a healthy natural ecosystem as "being 'stable and sustainable'; maintaining its organization and autonomy over time and its resilience to stress".

The definition of ecosystem death however is not as evident as expected in neither the information systems field nor the natural science field. The query on the research system data base (e.g. Google Scholar) provide a wide range of studies conducted in the natural ecosystem field about dying ecosystems, but no general definition of ecosystem death is provided. The closest definition we can find is the death of a specific ecosystem, for example, the tree death definition : "Death is defined as thermodynamic equilibrium between the organism and the environment, in which plants no longer have energy gradients to drive metabolism or regenerate" (McDowell, 2011).

First lets answer the question: what is death? Death is a defining feature of living organisms. In its most basic sense, death is the permanent termination of biological function within a living organism (Koshland, 2002).

Considering the previous two definitions, we propose the definition of software ecosystem death as: *" A permanent termination of an entity due to a disturbance in the dynamic between actors where collaborations and links are no longer occurring"*

We have learned from previous studies that the effect of actors and collaborations between them influence the overall health of an ecosystem (Jansen et al., 2009; Lungu, 2009; Bosch & Bosch-Sijtsema, 2010; Manikas & Hansen, 2013b), therefore, we propose with our definition of ecosystem death that their effect also influences the death of an ecosystem.

3.2 Software Ecosystem Health Framework

With an extensive literature review of the ecosystem health, Manikas and Hansen (2013a) looked at software ecosystems (SECO), business ecosystems (BECO), natural ecosystems, and open source software (OSS). They present in each category what guidelines fellow researchers used to measure their health. They finally propose a SECO health framework (figure 3.1), to measure the health of software ecosystems by assessing three main components: actors, software, and orchestration.

In the first component, a separation is made between two sub-components; first **individual actor health** represents the actors activity. The participation and engagement of an actor brings value to the ecosystem, and his robustness defines his stay in the ecosystem for a longer period. Second, **actor network health** which is the interaction between actors.

The second component is separated into three sub-components. First, **software component health** which represents the product in most cases. Its health is influenced by demand and quality. Second, **platform health** representing the technological platform of a software product. Its health is measured by measuring the effectiveness of applying the orchestration actions. Last, **software network health** which represents the interaction between the software components with other components.

Finally, the orchestration component revolves mainly around how the evaluation and monitoring of the ecosystem influences its health.

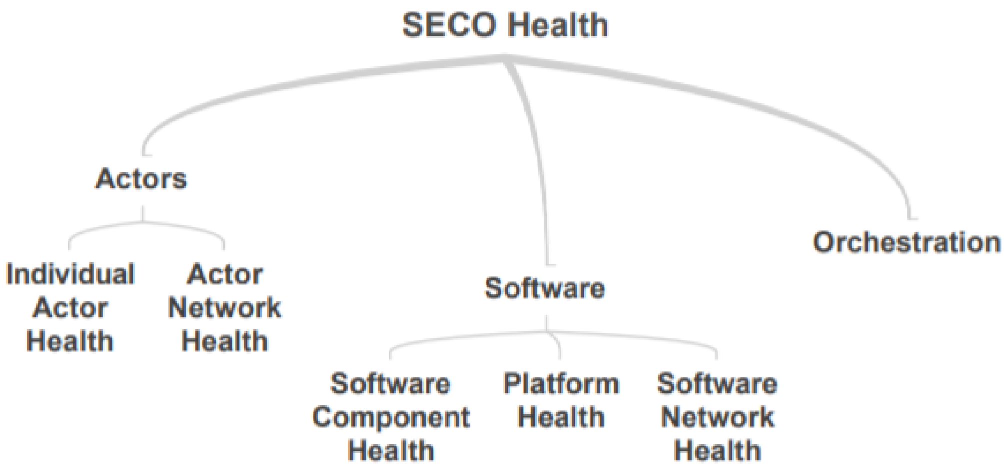


Figure 3.1: Representation of the SECO health framework breakdown (from Manikas & Hansen, 2013a)

3.3 Open Source Software Ecosystem Health Frameworks

Jansen (2014) points out the absence of operationalization method for measuring the health of open source ecosystems. Taking into account the three pillars of Iansiti and Levien (2004), the Open Source Ecosystem Health Operationalization (OSEHO) which establishes the health of open source software ecosystems is provided.

The pillars are separated in three areas; theory, network level and project level (see figure 3.2). The theory level describes the guidelines for the operationalization of the health concept inspired from the natural ecosystem. The network level translates the operationalization to the open source domain. Finally, the project level provides an overview of the metrics to describe ecosystem health.

Franco-Bedoya, Ameller, Costal and Franch (2014) created the QuESo quality model (figure 3.3) to measure software ecosystems built around an open source software (OSS). The model is composed of two types of interrelated elements: quality characteristics and measures. The quality characteristics are organized into three dimensions: **the platform** around which the ecosystem is built, **the community** of the ecosystem, and **the network** around the ecosystem such as projects or companies.

As a consequence of not finding measures for evaluating open source platform-related quality characteristics in their literature review, the authors decided to omit this part in their paper.

The community-related quality characteristics are defined by following a list of measures for maintenance capacity, process maturity and sustainability. First, they refine the maintenance capacity characteristics to three sub-characteristics: size, cohesion and activeness. The size refers

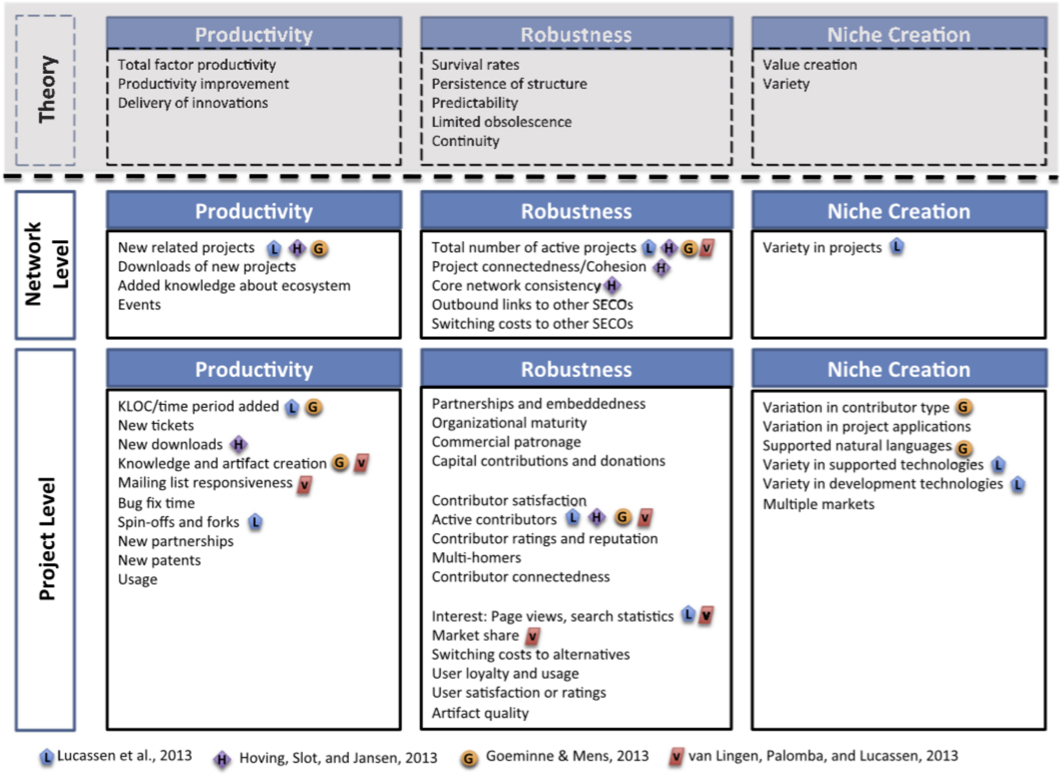


Figure 3.2: Representation of the OSEHO with occurrences of the metrics in the evaluated projects (from Jansen, 2014)

to the size of the community in number of developers or committees. Cohesion relates to the ability of the community to collaborate and connect with each other. The activeness relates to the activity within the community such as bug tracking activity and number of commits. Then, sustainability refers to the ability to maintain the product over an extended period of time. They again refine these characteristics to five sub-characteristics: heterogeneity, regeneration ability, effort balance, expertise balance and visibility. Heterogeneity revolves around contributors affiliation and community composition. Regeneration ability represents the growth of the community (e.g new members). Effort balance is about the uniformity of the contribution of community members. Similarly, expertise balance refers to the focus of expertise on a single contributor or small group of members. Visibility is the capacity of attracting people to contribute and support. Last, process maturity is the capacity for a developers community to follow process and achieve goals. However, no measures have been identified by the authors.

The ecosystem network quality is defined by two characteristics: resource health and network health. First the resource health characteristics representing the functioning and interaction of actors as close unit via the exchange of information. It is divided into two sub-characteristics; trustworthiness which represents establishing a trusted partnership of shared responsibility and financial vitality which represents viability and the ability to expand. Second the network health characteristics is divided into four sub-characteristics: interrelatedness as the connection between nodes; clustering as the classification of nodes around projects; synergistic evolution which is the dynamic and stable structure of subsystems within an ecosystem; and finally information consistency which is the consistency of the core information within the ecosystem.

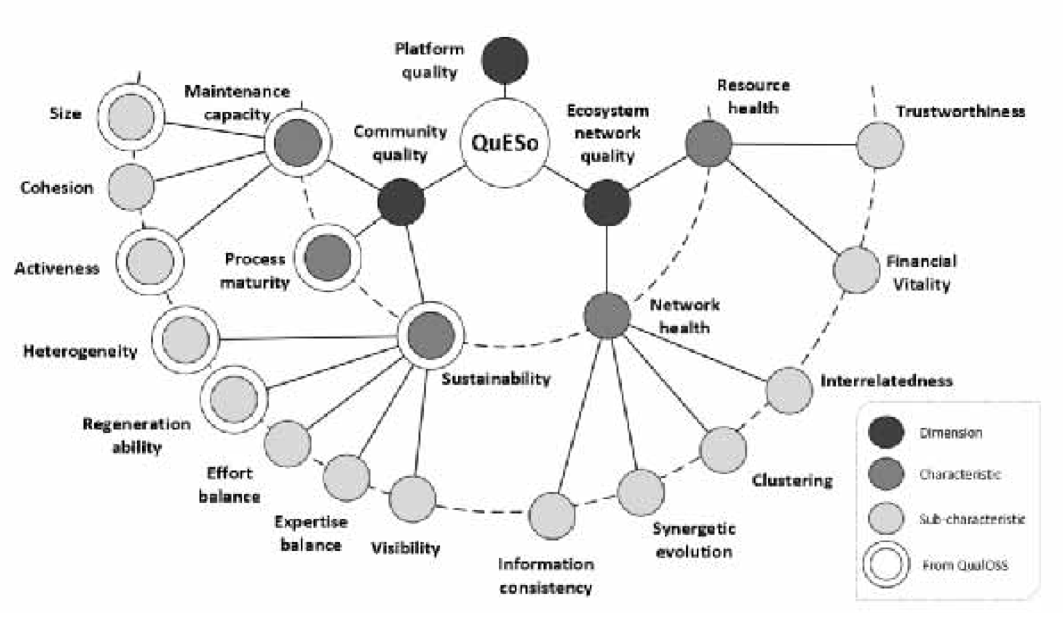


Figure 3.3: QuESo quality model (from Franco-Bedoya et al., 2014)

3.4 Business Ecosystem Health Framework

den Hartigh, Tol and Visscher (2006), create operational measurements using the three components of Iansiti and Levien in the business ecosystem level and in the company level. They make a selection of measurements out of a wide list using four criteria based on the goal of their research. The criteria of the measurements are: **user friendliness and understandability**, which allows managers to easily understand the model; **availability of data** in existing and accessible databases without the need of data mining; **long term usage** where managers are able to track development of business ecosystem health over time; and **company level measurement possibilities** where the measurements should be applicable for individual companies, for cross-sections of ecosystems and for ecosystems as a whole, which is also aligned with the first two criteria. Their concept initiates from the pillars: productivity, robustness and niche creation and evolves into **partner health** and **network health** after the selection of measurements (figure 3.4).

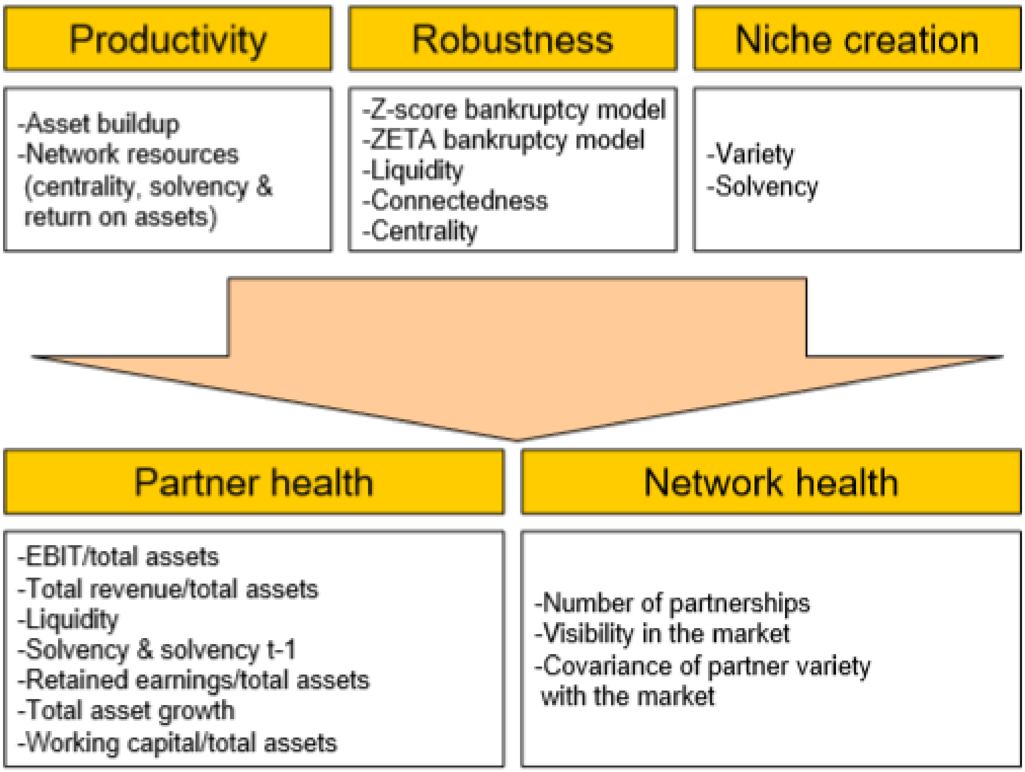


Figure 3.4: Business ecosystem health concept (from den Hartigh et al., 2006)

3.5 Comparative Study of Health Frameworks

With our research, we identify what can threaten the health of an ecosystem and point out the weaknesses and bad habits that are commonly practiced by dying or dead ecosystems. Establishing the health of an ecosystem cannot provide enough in-depth information on what part of that ecosystem could contaminate it's health. Therefore, we are aiming to detect and analyze links and practices that create the contamination.

Table 3.1 provides an overview on all the frameworks discussed in this chapter. With each framework, we present the aim of the research on respective papers, the source inspiration or pillar of the respective frameworks, the method used for the creation of the framework, the challenges encountered during their respective projects, how the framework was evaluated and finally the results brought by the framework.

Title	**The health measurement of a business ecosystem**	**Reviewing the health of software ecosystems: a conceptual framework proposal**	**Measuring the health of open source software ecosystems: beyond the scope of project health**	**QuESo a quality model for open source software ecosystems**
Publication	(den Hartigh et al., 2006)	(Manikas & Hansen, 2013a)	(Jansen, 2014)	(Franco-Bedoya et al., 2014)
Framework name	No name apart from health measurements	The SECO health framework	OSEHO	QuESo
Aim	To monitor the financial and network health of a business ecosystem	To create a discussion on the particularities of SECO health and define the health of software ecosystems	To evaluate the health of an ecosystem, identify weaknesses, make an ecosystem healthier	To assess software ecosystem quality considering the business oriented perspective together with the technical and social perspectives
Target	Business ecosystem	Software ecosystems	Open Source software projects	Open Source software ecosystems
Source	The classification from (Iansiti & Levien, 2004)	A wide literature review	Fundamentals basis from the works of (Crowston, Howison & Annabi, 2006) and (den Hartigh et al., 2006) used as pillars	Combines steps from (Radulovic & Garcia-Castro, 2011) and (Soto & Ciolkowski, 2009) for an OSS ecosystem scope.
Method	Design science	Systematic literature review	Design research	Systematic literature review
Challenges	The dependencies from Meso and Macro level is excluded, no distinction between different partnerships, restricted to Dutch ICT sector, missing customer data	Does not go into detail in the different kind of actors, the influences between entities is omitted	Missing data, lack of historical data	Missing measures
Evaluation	Analysis on ecosystems, partners and individual companies in the Dutch IT industry	No evaluation was conducted	Examining four research projects selected from literature surveys [(Hoving, Slot & Jansen, 2013),(Lucassen et al., 2013), (Goeminne & Mens, 2013), (Van Lingen, Palomba & Lucassen, 2013)]	Evaluation of GNOME using the QuESo characteristics with omitting the unavailable measures (maturity and resource health)
Results	Operational measures readily to be used by managers	A concepts of SECO health	A complete framework for open source ecosystem health assessment. The framework abstract from a project level and not from an ecosystem level.	A quality model with a bottom-up strategy (classifying measures)

Table 3.1: Literature review on software ecosystem health frameworks

Although each framework targets a specific type of ecosystem (i.e. software, business, open source), we are extracting all metrics discussed in different studies to find similarities between the different frameworks. Tables 3.2, 3.3 and 3.4 detail in each main pillar (productivity, robustness, niche player), the corresponding health metrics and which framework makes use of them.

Health metric	(den Hartigh et al., 2006) **model**	**SECO health** (Manikas & Hansen, 2013a)	**OSEHO** (Jansen, 2014)	**QuESo** (Franco-Bedoya et al., 2014)
New related projects		X	X	X
Download of new projects			X	X
Added Knowledge about ecosystems			X	
Events		X	X	X
KLOC/ Time period added			X	X
New tickets			X	
New downloads			X	X
Mailing list responsiveness		X	X	X
Bug fix		X	X	X
Spin-offs and forks		X	X	X
New partnerships	X	X	X	X
New patents			X	
Usage		X	X	
Tracked changes in projects		X		X
Job advertisements				X
Scientific publications				X
Liquidity	X			

First selected metric for Productivity: P1
Second selected metric for Productivity: P2
Third selected metric for Productivity: P3
Fourth selected metric for Productivity: P4

Table 3.2: Cross analysis of the **productivity** health metrics between the studied frameworks

We notice in the models by Franco-Bedoya et al. (2014) different measures are repeated in different categories, therefore we combine measures that target the same component but exist in different sub categories (e.g. number of contributors, contributors commit rate, and contributor activity graph: are interpreted as health metrics focusing on partnerships and contributor connectedness). Similarly, measures are repeated in the model by Manikas and Hansen (2013a). Therefore, the components have been interpreted with the best of our knowledge.

Health metric	(den Hartigh et al., 2006) model	SECO health (Manikas & Hansen, 2013a)	OSEHO (Jansen, 2014)	QuESo (Franco-Bedoya et al., 2014)
Existence of active projects		X	X	X
Project connectedness and cohesion			X	X
Core network consistency	X	X	X	X
Outbound links to other SECO's			X	X
Switching costs to other SECO's			X	
Partnerships and embeddedness	X		X	
Organizational maturity		X	X	
Commercial patronage			X	
Capital contributions and donations			X	
Contributor satisfaction			X	
Active contributors		X	X	X
Contributor ratings and reputation			X	X
Multi-homers		X	X	
Contributor connectedness		X	X	X
Interest: page view, search statistics		X	X	X
Market share	X	X	X	
Switching costs to alternatives			X	
User loyalty and usage		X	X	X
User satisfaction or rating		X	X	X
Artifact quality		X	X	
Number of members/ actors		X		X
Centrality	X			X
Build up of assets	X			X
Community efforts		X		X
Social media hits				X

First selected metric for Robustness: R1
Second selected metric for Robustness: R2
Third selected metric for Robustness: R3
Fourth selected metric for Robustness: R4

Table 3.3: Cross analysis of the **robustness** health metrics between the studied frameworks

Health metric	(den Hartigh et al., 2006) model	SECO health (Manikas & Hansen, 2013a)	OSEHO (Jansen, 2014)	QuESo (Franco-Bedoya et al., 2014)
Variety in projects			X	X
Variation in contributor type	X		X	X
Variation in project applications			X	X
Supported natural languages			X	X
Variety in supported technologies		X	X	X
Variety in developed technologies		X	X	X
Multiple markets	X	X	X	
Geographical distribution				X
Code vocabulary map				X
Solvency	X			

First selected metric for Niche creation: N1
Second selected metric for Niche creation: N2
Third selected metric for Niche creation: N3

Table 3.4: Cross analysis of the **niche creation** health metrics between the studied frameworks

Chapter 4

Case Study Design

4.1 Case Selection

Since a large majority of software practitioners publish in grey literature (i.e. blogs, white papers and web-pages), the use of multi-vocal literature review (MLR) which is a form of a SLR including grey literature in addition to the published (formal) literature (Garousi, Felderer & Mäntylä, 2016) is the best method for our case study selection.

The selection process starts by searching for articles from technology journals online stating "unhealthy" or "failed" IT projects, platforms and software ecosystems. To that end, we use the most common search engine (i.e. Google). After searching for "technologies that died ", the ten first web-pages that show as a result are selected (e.g. time.com or pcmag.com). An extensive list including different types of technologies is gathered, leading to a preliminary list of software ecosystems.

The extended list is then shortened and narrowed down to cases with the most potential to show patterns after data gathering. In order to narrow down our list we present some criteria. Those criteria should not necessarily require an extensive data collection since this practice is required for the next phase of the project: determining the health vulnerabilities.

- Criterion N°1: The case is an ecosystem on its own. Small ecosystems who are part of a larger ecosystem are also taken into account,
- Criterion N°2: The case is unhealthy and therefore fails the test of health metrics,
- Criterion N°3: Data must be available and can be collected from the needed websites (e.g. GitHub, Twitter platform,s website, forums, portals).

The cross analysis of the frameworks in the literature study chapter, provided us with a list of selected metrics. Those metrics were selected because they are commonly used in all four frameworks or at least three of them. We present the following set of metrics extracted from the 3 pillars:

Productivity

- P1: Updates
  This health metrics is a combination of "New related projects", "Bug fixes", "Spin-offs and forks" from the table 3.2, which makes us study all updates and changes in the cases
- P2: New partnership
  This health metrics targets new and consistent partnerships, to identify its consistency.
- P3: Responsiveness in forums and manuals
  This health metric corresponds to "mailing list responsiveness" from the table 3.2. For this particular metric we add the responsiveness in forums and manuals since it represents the same target group: active users and actors.

- P4: Events
  This health metrics studies the events organized to bring shareholders/actors together.

Robustness

- R1: Existence of active projects
  As the name of the metric implies, active projects provide a proof that an ecosystem is healthy.
- R2: Connected Network
  This health metric starts by measuring "active contributors " and combines "core network consistency" and "contributor connectedness".
- R3: Market share
- R4: User satisfaction and loyalty

Niche creation

- N1: Variation in contributor type
  Developers as well as partner are targeted since variety of contributors opens the door to a bigger range of users in the end.
- N2: Variety in support and development of different technologies
  We decide to combine both metrics about "support" and " development" since the same entity is being studied here: variety in use of technology. This metric however targets dependencies between projects, and the management system of the ecosystem, meaning that people with different sets of skills are being connected.
- N3: Multiple markets
  A case which is present in different markets uses different languages therefore a broader target group.

In table 4.1 we list the cases that passed the first criterion : the case is an ecosystem or is part of an ecosystem. This initial list of cases is displayed in the table with the marker selected or rejected after conducting a more thorough investigation and applying the remaining criteria (i.e. the selected health metrics and data availability). For more accuracy on our research goal, we focus the investigation on the years 2016 and 2017. Additionally, cases are rejected if there is proof of continuity with a new name or new concept (e.g. Visual Basic was turned into Basic Studio). Furthermore, a clear statement about the failure or discontinuity of a platform or a company is considered as a proof of death and thus the case is included it the research.

Case	Rejected or Selected	Reason(s)	Source
Alta Vista	Rejected	The company was bought by Yahoo! and the search engine was replaced by the yahoo search engine	Yahoo Official website [1]
AOL	Rejected	Expansion of the one platform follows P1 and R1	AOL official website[2]
		Proof of P2: New acquisitions of Paris-based startup AlephD and virtual reality studio RYOT. Merge with Yahoo to create new company called "Oath"	Tech Journal[3]
		"Oath" meets the metrics N1 and N2. Different contributors can be found. And it provides different technologies from news (Huffpost) to virtual reality (RYOT)	Oath's official website[4]
		Has a big market worldwide (N3) with over 1,700 employees in 18 countries around the globe	Oath's official website[5]
BlackBerry Phones	Selected	Considerable decrease in market share from 2007 to 2016	Market share statistics[6]
Deezer	Rejected	The service provided by the music streaming tool is availaible on diffrent platforms : Web, Android, FireOS, macOS, BlackBerry, iOS, Windows Phone and Symbian (Fulfills N1 and N2)	Official website
		The service is available on 187 countries (Fulfills N3)	Official website[7]
Ebay	Rejected	Increase in commercial benefits (e.g revenue growth of 8%). Steady annual revenue	Ebay Inc reports[8]
Eclipse IDE	Rejected	P1 fulfilled by presenting sable release and recent update	Official website[9]
		P3 is fulfilled: The forum is responsive and frequently up to date, Additionally there is a mailing list for every specific subject for a more optimal responsiveness. R1 and R2 are also answered by looking at the community responsiveness partially answered	Forum and news page[10]
		Events are held, and new ones are planned for the future	Official website Events page [11]
		The ecosystem is involved in several markets, this proof answers all the niche creation health metrics N1 N2 and N3	Marketplace page [12]
Firefox	Selected	Market share and download rate in decline	Market share study[13]
Google Picasa	Rejected	Retired and evolved to a new concept "Google photos"	Official website news statement [14]
Google Glass	Rejected	Increase in sale due to change of target costumers	Sale forecast[15]
Myspace	Rejected	No official advance since 2011 apart from the company being purchased in 2016 by Time inc. Unfortunately data is not available to properly study the case	Tech Journal[16]
PDF	Rejected	Unavailability of data to mine	
Pebble	Selected	Official statement of shutdown	Statement[17]
Symbian	Rejected	The last official Symbian phone was released in 2012. However there is no data available to properly study this case	Tech Journal[18], market share release[19]
Visual Basic	Rejected	Recent update: fulfills P1	Official website[20]
		P3, R1 and R2 are fulfilled by looking at the community responsiveness, and the blogs	Developer page community[21]
		The company offers compatibility with different platforms (i.e. Mac, Windows) and offers different features for a wider market N1, N2 and N3 are fulfilled	The official website[22]
Windows Phones	Selected	Official announcement for stopping production	Announcement[23]
Yahoo Search engine	Selected	Decrease in use and market share	Market share [24][25]

Table 4.1: Case Study Selection

Finally we detect five cases that have fulfilled the criteria to be selected as our case studies.

BlackBerry[26]:

In 1984, University of Waterloo engineering student Mike Lazaridis and University of Windsor engineering student Douglas Fregin found Research in Motion (RIM). RIM started developing communication products for military, police forces, firefighters and ambulance services in 1988. Few years later they began producing keyboard based devices with RIM 900 in 1996 and go public on the Toronto Stock Exchange in 1997. The very first device to carry the BlackBerry name was the BlackBerry 850, an email pager, released in January 19, 1999 followed by the release of multiple updated versions. 20 years of RIM were celebrated with the passing of one million subscribers in 2004. The next year the number of subscribers is quadrupled and the two founders, Balsillie and Lazaridis are named among Time magazine's 100 most influential people. In 2007, RIM hits a market capitalization surpassing $67 billion. However, the overwhelming response of Apple's new iphone, gives RIM a big challenge. The following year, its first touchscreen BlackBerry, is critically received comparably to the growing iPhone 3G model. BlackBerry sees Apple as its main competitor and launches its App World marketplace to compete with the Apple's App Store in 2009 but since then sales do not improve and the company goes through multiple executive decisions. In 2012, Lazaridis and Balsillie step down as co-CEOs and are replaced by Thorsten Heins who announces 5,000 layoffs and a delay to the BlackBerry 10 software update. A year later Heins finally unveils BlackBerry 10 and the first two smartphones to use the operating system, the Z10 and Q10 and officially changes the name of the company to BlackBerry. However, the phones don't sell well enough to revive the company leading Heins to announce another lay-off of 4,500 employees and being open to "strategic alternatives" including a possible sale. As a last attempt, in 2015, BlackBerry re-focused its business strategy and began to release Android-based smartphones, but since no improvement on sales have been noticed, the company announced it would cease designing new phones and rather focus on licensing to partners at the end of 2016[27].

The BlackBerry social media accounts are analyzed. Followers and hashtags are scanned for insights on user experiences. GitHub is used to look at collaboration projects and developers clusters. Finally, we focused on several general websites to have the market and business view of blackberry's journey over time.

Firefox:

Firefox was created in 2002 by the Mozilla community members who desired a standalone browser, and proved to be popular with testers who praised its speed, security, and add-ons compared to Internet Explorer. Following this positive test feedback, the browser was released in 2004 and started challenging the Internet Explorer dominance. In 2007 Mozilla started the relicensing project [28] from solely under the Mozilla Public License MPL which was considered by the free software foundation as a weak copyleft to a tri-license scheme of Mozilla Public License (MPL) with the GNU General Public License (GPL) and GNU Lesser General Public License (LGPL), leaving their developer with the choice of both licenses to work with. At the end of 2009, the usage of the browser grew at 32% making the version 3.5 the world's most popular browser, however it quickly declined in competition with Google Chrome A.1. A year later, IBM makes Firefox the default browser used by employees[29]. In 2012, Firefox become a free source code software[30], which permits anyone to view, modify, or redistribute the source code. As a result, several publicly released applications have been built from it, such as Netscape, Flock, Miro, GNU IceCat, Iceweasel, Songbird, Pale Moon, and Comodo IceDragon. Concerned that their performance is lapsing compared to Google chrome, Firefox initiated the Quantum project in 2016 which sought to improve Firefox's Gecko engine and other components in order to improve the browser's performance, modernize its architecture, and transition the browser to a multi-process model. Despite all efforts, Firefox's market share and the number of installs is in continuous decline compared to competitors.

Twitter gives insight on customer satisfaction and stories. The official website is analyzed for information about all the updates, the bug releases and the forums. GitHub is analyzed to study the collaborations between developers.

Pebble:

Pebble is the company behind the first commercially successful smartwatch. It was funded through a Kickstarter campaign which raised over $10 million between April and May 2012. Pebble started shipping their first watches to Kickstarters in January 2013, the watches can be connected to Android and iOS devices to show notifications and messages. An online app store was also implemented to distributes Pebble-compatible apps from many third party developers, including ESPN, Uber, Runkeeper, and GoPro. In 2014, a steel bodied variant to the original Pebble was introduced at the Consumer Electronics Show (CES) and the next year, the company launched its second generation of smartwatches: the Pebble Time and Time Steel. The devices were similarly funded through Kickstarter, raising $20.3 million from over 75,000 people. In 2016, Pebble cited financial issues after shutting down their subsequent Time 2 series and refunding users, leading to the official announcement in December the same year of shutting down the company and selling their intellectual property to Fitbit[31].

In this case we analyze the official website, blogs [1] and social media to discover the reasons that drove the company to shut down from the customer perspective.

Windows Phone[32]:

In 1996, Bill Gates Founder of Microsoft, launched Windows CE which laid the foundation for future mobile operating systems. Between 1997 and 2000, Microsoft released a couple of versions of Windows CE integrated in multiple updates of pocket PC versions which are hand-held computers with PDA (personal digital assistant) abilities. The introduction to Windows mobile started with the phone edition of pocket PC 2002 released in that year which included cell phone functionality in addition to the PDA abilities. Following this introduction, new versions of Windows mobile were released each year starting with pocket PC in 2003 to Windows mobile v6.5 in 2009, Windows Marketplace for Mobile, an app store, was also launched with this release. The next year, Microsoft launches Windows Phone 7 which was the only version of Windows Phone based on Windows CE, as future versions moved to the Windows NT kernel. In February 2011, at a press event in London, Microsoft CEO Steve Ballmer and Nokia CEO Stephen Elop announced a partnership between both companies declaring that Windows Phone would become the primary smartphone operating-system for Nokia, replacing Symbian[33]. Following the new partnership, Nokia Lumia operating with Windows 8 was introduced in 2012. The following year, Microsoft acquires Nokia's mobile phone division outright, and managers revealed that the acquisition was made because Nokia was driving the development of the Windows Phone platform to better match their products[34]. As a result, Nokia's hardware division became a subsidiary of Microsoft operating under the name Microsoft Mobile. In 2014, Nokia released the Nokia X series of smartphones, using a version of Android forked from the Android Open Source Project with a user interface modified to resemble Windows Phone's interface[35]. Additionally, Cortana, a Microsoft's voice activated digital assistant for voice-based search was introduced the same year. However the partnership came to an end after Microsoft sold Nokia in 2016, followed by a confirmation from Microsoft's corporate vice president, Joe Belfiore that Microsoft will no longer sell or manufacture new Windows 10 Mobile devices due to low market-share and lack of third-party development, but would continue providing bug fixes and security updates to existing devices in 2017[36].

The official website is analyzed for all the updates/upgrades and releases the ecosystem has gone through over the years. Social media websites give us insight on the users community and their opinions on the products. GitHub is studied to discover the cluster of developers that collaborated on projects. Other websites are analyzed for more marketing and commercial information.

Yahoo:

Yahoo is a web service provider founded in January 1994. It was one of the pioneers of the early Internet era in the 1990s and it slowly started to decline from the year 2000 as the company made one wrong decision after the other[37]. Yahoo had the opportunity to buy Google for $1 million in 1998 and declined. The opportunity came back in 2002 but Yahoo considered the $5 billion price Google was asking to be too high. Today, Google's parent company Alphabet is valued at over $500 billion. Instead, Yahoo acquired Broadcast.com in 1999 for $5.7 billion. In 2001, Hacker Adrian Lamo modifies various older Yahoo News stories, and points out security flaws[38]. In 2004, Yahoo announces the formation of Yahoo Research Labs, a research organization for the

[1] `https://developer.pebble.com/blog/`

invention of new technologies and solutions for Yahoo in addition to launching its own search engine technology. Between July 2004 and July 2005, Yahoo acquires multiple companies in line of email messaging and communication platforms such as: email provider Oddpost[39] and VoIP (Voice over IP) provider DialPad Communications[40] to announce in August 2005 their integrated DSL (Digital subscriber line) service[41]. The company's value dropped to approximately $45 billion in February 2008 when Microsoft offered to buy Yahoo's outstanding shares and thereby getting a hold of 61% stake in the company. However, it refused to take the offer, but 2 years later, Yahoo announces search alliance with Microsoft[42] making Bing it's organic research engine. In July 2012 Yahoo Voices was hacked, compromising approximately half a million email addresses and passwords [43]. And in 2016 the company announced a data breach of user account data that occurred sometime in late 2014, and affected over 500 million user accounts[44]. The announcement however came after Verizon agreed to purchase Yahoo's operating business for $4.8 billion in July 2016. The acquisition was completed in June 2017[45].

We focus on the company's websites to gather data on the extensions, releases and updates Yahoo platform offers. Forums and social websites are targeted to collect data on customer relations and user experiences.

4.2 Detection of Health Vulnerabilities

We theorized back in our introduction that failing ecosystems follow a pattern of health vulnerabilities leading to an unhealthy ecosystem. We are aiming to detect those patterns by means of data mining and qualitative content analysis. The health metrics previously identified, are used as an anchor to determine which sources and which aspects of the ecosystem should be studied in order to achieve our goal.

With the **productivity** pillar represents the growth of the company which is measured by the continuity of new projects and new partnerships. We have four metrics to consider in this pillar. First, we target the website of each case and define the timing for release dates, launch of new plugins, and to determine the type of fixes that have been made. Second, we target the official website along with tech blogs and journals to study acquisition of new ventures or new partnerships, in order to determine if it is harmful for the ecosystem. Third, it has already been stated that a healthy ecosystem is an ecosystem that informs it's actors and beneficiary well. We are targeting the official website along with the community blogs to discover if there is frequent updates in forums and manuals. Our theory is that certain practitioners do not inform the end user well due to lack of technical expertise and support. Fourth, we are looking at the logs of events organized for the product, platform or ecosystem in general. We aim to determine what are the common misfortunes encountered during those kind of events and how to avoid the pitfall.

The **robustness** pillar mainly refers to the capability of surviving disturbance. In this pillar the main focus is the connectedness of the actors and the community. Therefore, we are looking at collaborations, commits and forked projects within GitHub to define the size of the developer community and the frequency of collaboration between them. Additionally in blog communities and social media we are analyzing the user satisfaction and the impact of those channels on the ecosystem.

And finally the **niche creation** pillar details three metrics. This pillar refers to the diversity in the ecosystem that leads to innovation. In perspective, we are focusing on the ecosystem's market and choice of target group. Since choosing the right or wrong group of people or market to release your platform or product can make a difference for the ecosystem health. For example, we included Google glass as a failing project in our preliminary list. After further investigation, we have discovered that although the product have failed to meet the expectation of the public in its early years, it has been massively and successfully used in the private sector. It appears that what was viewed as an issue in the first released versions (i.e. privacy) does not matter to a target that make use of the glasses to operate on patients for instance (e.g. surgeons). We were aiming to study Google glasses as our first case but to our surprise it has already auto-corrected its path towards being more healthy.

As mentioned before, each health metric targets a specific data which is collected from a specific data source. However our biggest data source provider is the end-users gathering hub, which is mainly social media and forums. Table 4.2 provides an overview of each metric and the targeted data base which is mined for our findings. In the next chapter, we break down each type of data source .

Pillar	Health metric	Data mining target
Productivity	Updates	Official websites update pages
	New partnership	Official websites, commercial and news pages, articles
	Responsiveness in Forums and manuals	Official websites, forums and community blogs pages
	Events	Official website news and events pages, conferences and events websites, social media websites
Robustness	Existence of active projects	GitHub, official websites, news page
	Connected Network	Official websites, community pages, social media, GitHub
	Market share	Official website, commercial and financial pages, market share websites
	User satisfaction and loyalty	Forums, social media websites, community blogs
Niche creation	Variation in contributor type	commercial pages, news pages
	Variety in support and development of different technologies	Social media website, official website
	Multiple markets	Official website, news pages

Table 4.2: Targeted websites for data mining

Chapter 5

Data Collection

For each case study, data have been collected from multiple sources, including social media websites, users forums, developers forums and many more. In the following sections we specify the source used for the targeted data and the specific data collection methods used in each source.

5.1 Source 1: Forums

As mentioned earlier, the productivity of an ecosystem can be evaluated by the responsiveness in forums. Additionally, customers reactions in forums and community blogs give information on the robustness of the ecosystem in terms of connectivity of the network and user satisfaction. Pitta and Fowler (2005) stated that the implication of forum use can help marketeers design new products while take advantage of the direct source of wants and needs of customers. For our research, we are collecting the subjects discussed by customers and developers on forums to find out if the studied platform is taking into count this useful tool for their marketing strategies.

We extract subjects, date of creation, number of views and number of replies from the forums in order to have a time chart linking discussed subjects to external or internal changes in the ecosystem and to study the time of response. Since each website and webpage has a different HTML script, our algorithm is tailored to each one of them. The HTML text is first extracted and each element we are collecting is cut from the text and stored in a comma-separated value files (CSV).

Websites are locally downloaded for an optimal crawl of the requested data using the spider crawler technique with a C# code. A crawler is a program that automatically explores the World Wide Web by retrieving a document and recursively retrieving some or all the documents that are linked to it. The crawler visits web sites and reads their pages and other information in order to create entries for a search engine index. Spiders are called spiders because they usually visit many sites in parallel at the same time, their “legs” spanning a large area of the “web.” (Kraft & Myllymaki, 2009).

All forums data have been collected as of March 6th, 2018 with some limitations; we could not collect forum data from dynamic websites using web crawlers, thus only static websites have been used to this end. Dynamic websites protect their users and users data by denying access to multiple requests that have a time span shorter than 2 seconds between them. Hence, any web crawling technique is automatically blocked by the websites security.

5.2 Source 2: Social Media

Since the rise of social media websites, researchers and professionals have been using the big flux of data to analyze users sentiment and opinion in a wide range of topics: political opinion (Ceron, Curini, Iacus & Porro, 2014), social behavior (Stieglitz & Dang-Xuan, 2013) and customer perspective on the market place (Sarlan, Nadam & Basri, 2014).

We decided to focus on Twitter as our main social media source, since it is the most popular micro-blogging platform in research (Deng, Sinha & Zhao, 2017). Twitter is a service allowing people to easily communicate in a free, rapid way. The social media has a 140-characters text allocated area that would force the users to convey their message in a clear and direct way. Unlike other micro blogging platform (e.g. Facebook, google +) who give freedom to the user to have a long text without restriction, which might make the post discursive and unclear.

Iansiti and Levien (2004, p. 5) stated that the terminology of ecosystem "can help focus managerial attention on features of modern business networks that are often ignored by conventional theories about markets and industry structure that underlie many drivers of business success and failure". From our interpretation of the statement, we decide to narrow down our data to essential information, which is the reason of death and failure of the platform we are studying. Furthermore, the term "failure" is widely known and used among the public, compared to the term "unhealthy ecosystem". Therefore, since Twitter users make use of hashtags to mark the visibility of their tweets and highlight certain topics (Agarwal, Xie, Vovsha, Rambow & Passonneau, 2011), we are collecting tweets using the hashtag "#fail" in addition to the name of the chosen ecosystem.

In order to collect data from Twitter, we are required to create a sample application. The application creates OAuth access codes which is a standard identification mechanism. OAuth credentials are then used to make requests to the social media's Application Programming Interface (API). OAuth stands for "open authorization" and provides a means for users to authorize an application to access their account data through an API without the users needing to hand over sensitive credentials such as a username and password combination (Russell, 2013, p. 403).

Twitter is crawled with a spider crawling technique using Python code. We enter the parameters "–Queryserch", which represents the hashtags we are looking for, and "–maxtweet" which represents the maximum number of tweets to be collected. The corresponding data is then stored in CSV files with columns including the username, date of publication, and most importantly the text of the tweet along with the targeted hashtag.

All twitter data have been collected as of January 17, 2018. As it usually happens with data collection, some limitations have been encountered. The maximum tweet request for our Python code during the tweets collection could not exceed 10000. While testing with a greater number the kernel shuts down and does not return any results.

5.3 Source 3: Developer Network

As defined by Joblin, Apel, Hunsen and Mauerer (2017) a developer network is a relational abstraction that represents developers as nodes and relationships between developers as edges. The network usually represents a platform where third-party developers can communicate and exchange knowledge. Contributions can vary from threads and comment in developer portals or source code files and commits in projects.

Due to the importance of the developer community in the ecosystem health and their connectivity (Manikas and Hansen (2013a); den Hartigh et al. (2006); Jansen (2014); Franco-Bedoya et al. (2014)), we analyzed the developer community to extract any health threats that might occur in this aspect of the ecosystem.

We collected data from the developers portals of the 5 case studies in order to determine the activeness and connectivity of the community by analyzing the response time and the amount of subjects discussed. Additionally, we generate network social graphs which is a visual representation of all participants on the repositories provided by the official Github profile of our case studies.

5.4 Source 4: Miscellaneous Sources

In our data analysis, we are integrating online market share releases, official article releases from the targeted cases, and grey literature representation in online web news. The side-back faced with collecting data from those online sources was the use of dynamic websites. Unfortunately,

Data Source	Collection method	Final retrieval time	Number of hits	Target group
Twitter	Spider crawling with Python and Tweepy module	January, 2018	~15,000 tweets	End Users
Forums	Spider crawling with c# (individually tailored to each forum)	March, 2018	+650,000 posts	Customers, Developers
Github	Python's PyGraphviz module and Github-Social-Graph package	May, 2018	+4,000 repositories, +4,000 collaborators	Developers
Market share releases, press conferences, journals	Manual keyword search	July, 2018	+40 articles	Business

Table 5.1: Summary of data collection sources

dynamic websites do not allow the scraping and crawling technique. Therefore, those sources are not mined in the same scale as the previous ones, and the data is not analyzed along with the forums and the social media data, but rather as a connector between the business view of the ecosystem and the findings from sources 1, 2 and 3.

In conclusion, this chapter reviewed the methods used to collect the data and establish the visualization of the developers network. Table 5.1 and figure 5.1 summarize the important points from previous sections. In the next chapter, we discuss the findings by diving into the data and analyzing its content.

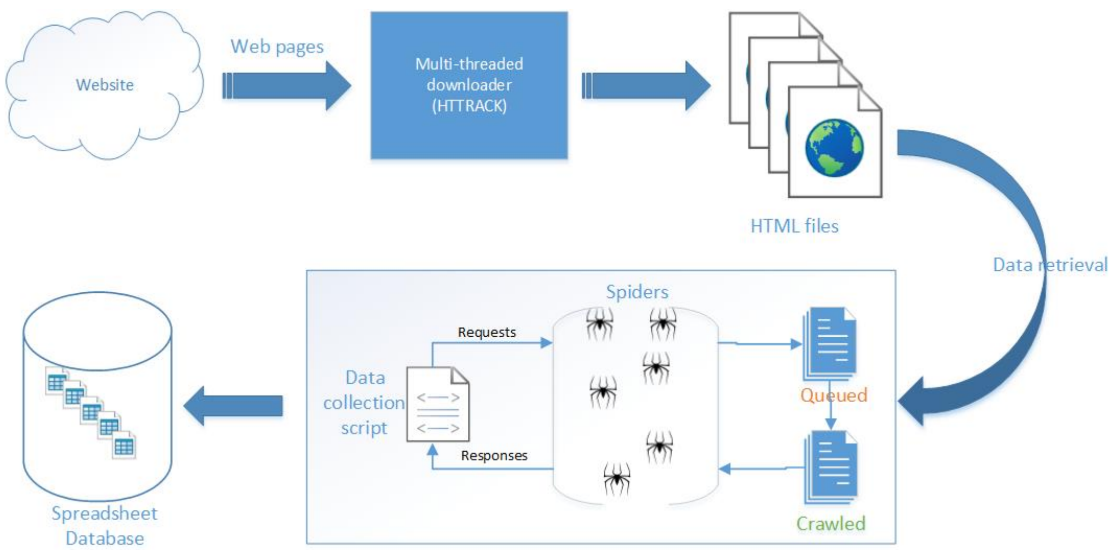


Figure 5.1: Spider crawler architecture, inspired by Castillo, 2005

Chapter 6

Identifying Health Vulnerabilities

6.1 Data Interpretation

This research focuses on the threats that an ecosystem can face in terms of market level, customer level and technical level, since these levels are the main sectors that face challenges when introducing software ecosystems (Valença, Alves, Heimann, Jansen & Brinkkemper, 2014). We identified eight nodes in the collected data.

1. **Customer care:** refers to customer satisfaction and customer service. As an example, when users are mentioning updates they have not been notified of by email or else, and are asking for its legitimacy on the forum. Some of the recurring subject involve: users specifically complaining about the frequently asked questions (FAQs) and the customer care workers, and asking for personal assistance.

2. **Tutorial:** all "how to " questions have been indicated in this section. This node shows that users have not been given the appropriate information to deal with the product/ software.

3. **User interface:** This node is about all interface functionality that are not working or missing . For example: missing buttons.

4. **Brand switching:** refers to the change from the current studied ecosystem to a competitor. Either it is explicitly mentioned that the customer changes sides, or the user has asked for reviews or advice from other customers about a competing platform.

5. **Incompatibility:** refers to external entities which are not able to function in the platform proving a poor relationship between ecosystem nodes. For example: websites not loading on Firefox, installation and add-ons issue specifically mentioned for certain OS or lack of compatible application for a mobile phone.

6. **Bugs and crashes:** highlights subjects explicitly referring to a freeze, a crash or a bug and problems mentioned after updates.

7. **Requirement:** contains problems encountered when basic actions and requirements are not fulfilled within the platform. For example: download of the software, opening emails, search results, difficulty of syncing.

8. **Security:** contains all security issues referred to by the user regarding malware, safety of websites, login issues of emails and unplanned deletion of content such as emails and documents from the user's account or device without their consent.

In the data analysis phase, all subject nodes are organized by year and by case study. The result give us an insight on what issue are more dominant than others during certain moments of the time line. We notice that certain subjects are frequently mentioned on specific dates and

Node \ Case	BlackBerry	Firefox	Pebble	Windows phone	Yahoo	Total
Brand Switching	15	2,532	18	191	1,534	4,290
Crash and Bugs	219	14,598	5	584	44	15,450
Customer Care	5,591	12,434	7	1,577	6,146	25,755
Incompatibility	5,716	8,897	37	10,582	197	25,429
Requirements	2,065	2,537	10	1,056	246	5,914
Security	1,779	10,765	1	542	13,023	26,110
Tutorial	475	7,473	3	806	648	9,405
User interface	1,389	15,158	9	1,324	133	18,013

Table 6.1: Coded data matrix

theorize that those times are often related to an actual event happening in the ecosystem (e.g. a merger, a software update). The number of data coded in each node is referred in the future as data coverage. Data coverage determines a percentage of the mention of a certain subject compared to the overall data available in our system.

Table 6.1 represents the results on the number of nodes detected by the automatic coding pattern algorithm in each case. It is important to note that data collected for certain cases is greater than others due to differences in data availability and lifespan per ecosystem. For example, Pebble's forum only started from 2013 and stopped being in use by 2017 whereas Firefox's forums start from 2006 and continue in 2018.

Figures 6.1 and 6.2 represent the data coverage of nodes detected through the content analysis phase in all 5 cases combined. With the visual help of the charts, we can retrace specific quotes in the time line and in a specific data set to support our results. Furthermore, individual data chart per case can be found in appendix A.

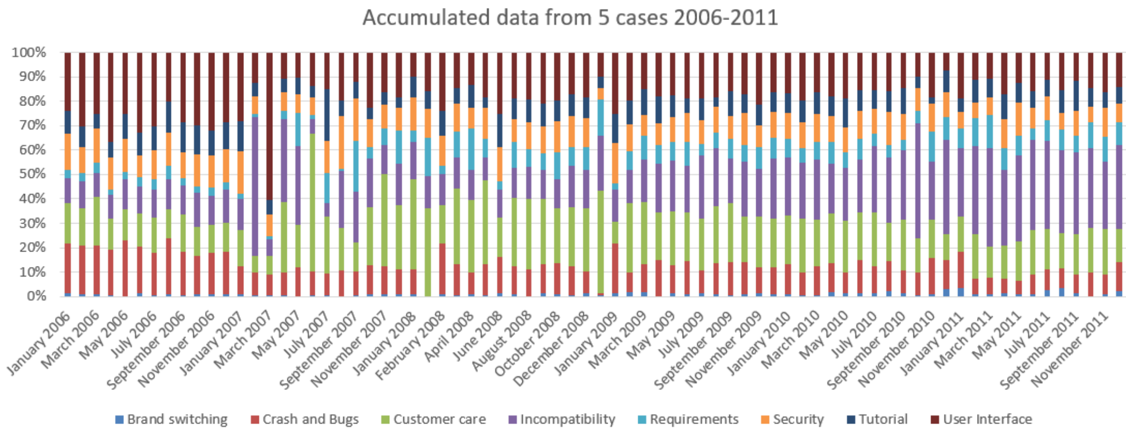


Figure 6.1: Data chart representation of identified nodes in all 5 cases combined (1/2)

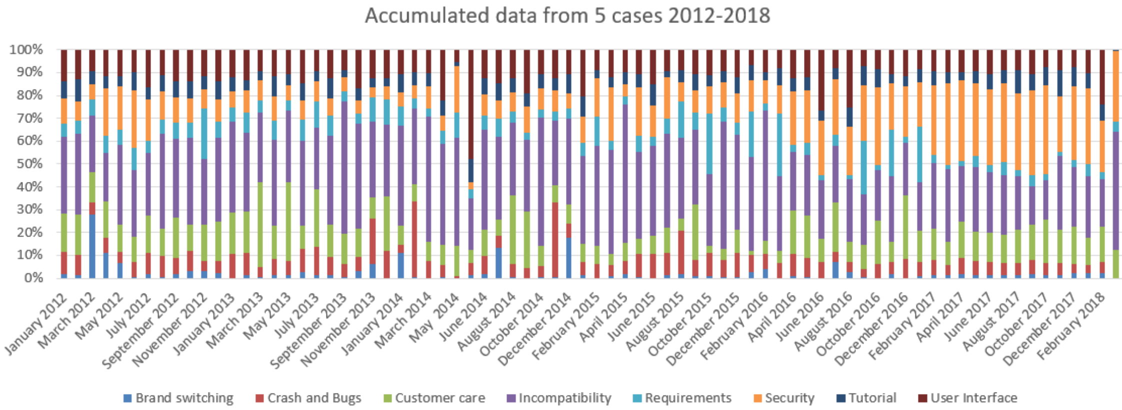


Figure 6.2: Data chart representation of identified nodes in all 5 cases combined (2/2)

6.2 Health Vulnerabilities

We derive the following health vulnerabilities from studying the nodes and their implications, we extracted the threats that score the highest and are in our knowledge the most important to deal with in an ecosystem's health. Furthermore, we specify which source have made the most impact in concluding those health vulnerabilities which can be based on customer feedback, business observations or a combination of both.

6.2.1 Insufficient Partner Collaboration

As suspected at the start of the research, the data from the Windows phone case (A.2) clearly confirms our hypothesis on the incompatibility issues faced by the platform. The incompatibility node is dominating all the data, with 50% to 100% of coverage through the time line. Similarly with the Blackberry case (A.3), the incompatibility subject is constantly mentioned throughout the whole time line, although it is mostly dominating over 50% of the data from mid 2013.

The commonly flagged subjects in Windows phone forum feeds and discussions include rating and ranking certain applications (apps), developers presenting their own apps to the community, or individuals sharing "unofficial" apps outsourced and created by third parties instead of the official publisher (e.g. the PokemonGo app and the Facebook mobile app). Developers face annoyances with *"store in-app purchase breakage"* and *"errors after publishing app to the store"* and users are unable to install or use certain applications after updating their operating system (OS); *"Angular app stopped working with latest OS update", "Why am I unable to install apps on my Lumia 520 after updating it to Windows phone 8.1?"*. Additionally, the lack of certain commonly used applications is a constant occurrence; *"decided I wanted to try out Instagram, then realized I don't own any devices that are compatible", "Microsoft: Google Is (Still) Blocking Us From Building YouTube for Windows Phone"*.

Unsurprisingly, Blackberry and Windows phone incompatibility is facing the same type of issues concerning fake and unofficial applications that do not have the right security clearance or even the right functionality; *"terrible Facebook app on blackberry 10","Nearly downloaded the fake BBM app for android there"*. Moreover, Blackberry users and developers express different opinions when it comes to the decision made in summer 2013 of making BlackBerry messenger (BBM) cross-platform which then allows Android, iOS and Blackberry phone users to enjoy all features of the chatting app without restrictions. Cross-platform applications encourage the presence of multi-homers which is a sign of robustness in ecosystem health (Jansen, 2014; Franco-Bedoya et al., 2014; Manikas & Hansen, 2013a). Although, it appears that actors in the ecosystem, precisely users, don't agree; *"BlackBerry is killing itself by making BBM cross platform"* and others point out yet another highlight on how other platforms are supporting the very own Blackberry technology better than Blackberry itself; *"It seems like BBM on Android is working far better than BBM on my BlackBerry"*. In contrast, developers were pleased when Blackberry decided to introduce phones supporting Android applications; *"Installing Android apps on BlackBerry 10 just got easier - All aboard the APK Train"*.

Pebble's incompatibility scores 24.67% of the data coverage. The main mentioned issue is yet again the lack of applications, the inability to install more than 8 apps, and device incompatibility with iOS; *"Pebble cannot work reliably on iOS", "Can it work reliably on iOS? No, it can't." "Pebble focus on getting things working reliably on iOS7! At least give some info about upcoming updates, it's been an eternity"*.

Firefox has a very low score on incompatibility with less than 6% coverage. Which refers to 3 peaks in the time line chart on September 2012, May 2015 and May 2016. The mainly mentioned experience is the incompatibility with certain operating systems; *"new problem today On Windows 10", "Problems migrating from Windows 8.1 to new computer running Windows 10","Firefox have a performance issue on Linux"* and other functionality provided by other platforms; *"Firefox isn't getting along with AdWords or adCenter"* which are respectively provided by Google and Microsoft.

Lastly, Yahoo scores the lowest score in all our cases with 0.43% of all coverage of incompatibility. The issue has only been mentioned in November 2015. The users are facing email restrictions when trying to open it on a mobile app. The issue was occurring on Android phones, Windows phone and iPhones. Furthermore, Multiple subjects in the forum refer to the the change of domain name; *"Change yesterday from co.uk to com","How to change the @yahoo.com or .ca", ".com / .co.uk conflict"*. This change of protocol might have been the reason relating to issues for users to log into their emails, but due to time constraint we cannot prove this hypothesis.

In conclusion, we named our first health vulnerability insufficient partner collaboration because all our cases register a lack of partnership when it comes to cross-platform collaborations. This health vulnerabilities have been derived solely from studying the opinion of the customers and developers.

6.2.2 Lack of Platform Security

The security node has a high score after two major security breaches where 3 billion yahoo accounts have been hacked[1] and it seems that customers no longer trust the platform because of it. They then turn to forums for confirmation about legitimacy of messages received by supposedly yahoo agents *"Repeat warnings from yahoo or imposter party", "Is this a legit Yahoo message?"*.

While the security breach is vastly present in the Yahoo case in comparison to the others, it also has a high importance in the Firefox case. The threat is mainly discussed in the last year (January 2017 to January 2018). Almost all forum subjects refer to a Secure Sockets Layer (SSL) certificate error which appears to happen only on Firefox and not on other browsers *"Getting an SSL Certificate error on Firefox after turning on DPI SSL on our SonicWALL. When I use Edge or Internet Explorer, the issue doesn't happen.", "Firefox is the only browser not recognizing my SSL certificate.", "Firefox rejects most https sites."*. However, the solution to this issue was presented by one of the top 10 contributors in the Mozilla forum, and users had to manually export Zscaler[46] certification from Internet Explorer or Chrome and import it to the Firefox certificate paths[47].

Furthermore, Blackberry's security scores over 10% the whole time but the score gets higher than 35% between February 2015 and November 2017. The time reference is understandably relevant to the fact that the data is pointing to the 2016 BlackBerry security summit. The event showcases the collaboration of corporations regarding company security, a live hack on stage to demonstrate the security threat that internet of things faces and more[2].

In conclusion, we consider security to be a very important health vulnerability for an ecosystem, since it leads to future partners preferring to work with a protected platform rather than a

[1] https://techcrunch.com/2017/11/08/yahoo-senate-commerce-hearing-russia-3-billion-hack/

[2] http://blogs.blackberry.com/2016/07/blackberry-security-summit-2016-recap-customer-wins-giuliani-keynote-hacking-by-coffee-pots-and-more-video-pics/

vulnerable one. This health vulnerability was visible in the combined study of multiple targets and sources: users, developers, business activities, and public events.

6.2.3 Lack of Internal and External Quality

This health vulnerability represents the nodes touching the quality of the platforms and their products, customer-specific complaints, missing features and unwanted updates (i.e. the nodes crashes and bugs, requirements, and user interface).

Windows phone's requirements node scores 10% to 30% between November 2010 and November 2014. The biggest issue faced by users here is regarding battery usage and its overheat; *"battery getting too hot in Lumia 520 and 720", "Lumia 920 post-Windows Phone 8.1 battery life sucks!!","Fixed my 920 battery drain.","Battery life back to normal?","Battery heating problem fix"*. This data shows the importance of choosing the right manufacturers to associate with and add to your ecosystem.

In Firefox, the user interface node is dominating the data coverage with a score up to 38%. Some subjects refer to the disappearance of buttons and addons *"I can't find the New Tab Preferences under the gear icon.","Addon buttons disappeared can't find a way to put them back","Where did the button to add calender entry for .ics attachments go?","Where did the Home button go in Firefox 57 (Quantum)?"*. Others mention the appearance of unwanted buttons *"The new update with the blue button moving back and forth is maddening.", "how do I get rid of new QR button - or go back to old version"*. Lastly, customers experience the misfortune of common features ceasing to work as usual *"Firefox loads fine and I can get to websites but whenever I click the menu button in the top right it hangs and I have to kill the process", "The find feature doesn't work correctly when it tries to find a phrase."*.

The qualitative analysis of the subjects align with our own data collection experience. During the data mining of the forum websites, we faced side backs because of inconsistency in the source code of the forums. With this personal experience we observe that the Firefox's web developers implement their code in an unstructured and unclear manner, providing a bad internal quality of the platform.

Furthermore, bugs and crashes seem to be frequent among Firefox users complaints including syncing issues. Most of them report the problem without even knowing the source *"after installation I only get 404 not found","FF 57.0 and 57.0B14 both crashed on startup", "Firefox crashes - Even after un-install - re-install"*. Others, identify its origin to a recent version update *"website https://webqr.com stopped reading QR after Firefox updated to 58.0","Since I installed update Firefox is not working properly ", "updated to V58 and now new tabs doesn't open"*.

In addition to email technicalities (*"Unable to deliver message bounce email", "Not receiving any emails to my inbox"*), Yahoo sports platform presented a lagging stream incapacitating users to watch their game: *"Only reason I use yahoo is for fantasy football and their app is not working"*. User interface complaints was also present in Yahoo about non functional buttons: *"Sign Out Button is Not Working", "Unresponsive Icons and Buttons", "cannot change password. cannot find button"*. And as also seen with the other cases, changes of interface is not accepted by users since the target group have not been consulted in the matter before the change: *"Your new mail interface is a mess!! It isn't intuitive & almost impossible to navigate!"*.

Pebble's data indicate that predefined requirement have not been met, meaning the battery life and the waterproof features *"supposed to be waterproof but then it isn't","Pebble Battery Drain on Phone"*, resulting in customer dissatisfaction *"that's not what was promised"*. Additionally, in terms of external partners contributing to the problem, we surprisingly register a high amount of subjects about delivery issues, for example, long delivery time is mentioned by customers, the package never arriving to destination, or receiving a malfunctioning device on delivery. Even if the matter, internal or external, has been reported it doesn't get fixed for a long period *"they still can't get this damn issue fixed"*. The user interface node refers mainly to issues regarding button customization, the watch display and the notification management of the watch on the connected smartphone. Furthermore, in relation to all complains stated earlier the device keeps on crashing *"Phone crashes immediately at bootup when I got pebble app", "Pebble app crashing"*.

Concluding the analyzed data for this health vulnerability, we found that updates often fail and features are often missing or not functional. Additionally, the choice of your component provider, delivery service and your development team seems to be important and can jeopardized your ecosystem. However, due to time constraint this health vulnerability has only been studied from the customer perspective and should be subject to further study in the future.

6.2.4 Customer Negligence

In this health vulnerability we combine the results from the customer care node and the tutorial node, since they are both represent the essence of this actor. The customer care concept is revolving around the user satisfaction aspect, their loyalty towards the ecosystem, and at last the way they are managed and how the ecosystem interacts with them. "Low customer satisfaction rates can be devastating to an ecosystem ... If not satisfied, customers can, and will, change to a competitor" (Van Den Berk et al., 2010, p. 131). And that is the reason why we consider customer care as an important link between the end user and the organization. Our data supports Van Den Berk et al. remark and our previous statement, since customers often refer to the incompetency or and unresponsiveness customer care a motive for them to switch to a competitor ecosystem.

Our first look at the data shows that the forums provide support pages for issues faced by users when using certain products. By following certain steps, the problem can be resolved by the users themselves. Furthermore, contributors exist in the community, but the rate of opening and solving subjects is not fast enough (i.e. taking days sometimes months instead of minutes or hours).

A more extensive analysis of the data shows that, in Pebble for example, the communication between users and customer service is reportedly poor when it comes to listening to the customer wishes and needs. Customers end up not agreeing with the corporation decision making: *".... New smartwatch... So no update to the original pebble?", "Pebble you need to work on existing customer dissatisfaction before trying to lure in new ones ","Pebble $150.00+ for a product that doesn't work!!?? Where's the product support team?"*.

In general, our interpretation of the data concludes that customers have the means to ask questions via forums. However, for a faster response or for a subject of big magnitude (i.e. explanation on a big hacking issue that have not been publicly communicated) the customer care lacks in effectiveness and anticipated reactions and the most faced outcome by users is ignorance and negligence.

In all research definitions of software ecosystem, actors are defined to be the entity that gives an addition to the platform. Usually those actors are considered to be developers creating components to the platform, a partner making the technology more complete or a vendor making the ecosystem bigger and with higher reach. The debate continues as whether to include customers as a variation of ecosystem actors or not. Our research points out that the customers are often neglected and their view seem to not matter as long as the business model is approved by a more prominently positioned actors in the ecosystem, when in reality without customer to sell the product to, the ecosystem wouldn't exist and grow.

Finally, this health vulnerability, was derived mainly from customer's testimonies, but also with the analysis of responses related to customer complaints.

6.2.5 Insufficient Competitiveness

We deduct this final health vulnerability from underlying subjects always present in the analysis of all data sources. Our forum and social media data showed that users usually compare our case studies to their competitors and don't hesitate to switch if their needs are not fulfilled. In Pebble's data for example, the customer dissatisfaction due to functionality issues or incompatibility lead users to switch to a different brand *"had to bring out my old casio to go for a run", "Guess I'm getting a Moto 360"*.

Our five cases touch a variety of sectors: mobile phones, search engine, browser, smartwatch. However, we found out that they all have poor strategic skills, for example, using licensing business model instead of an open free source (Microsoft's Windows against Google's android) or creating a new product line instead of fixing consistent update issues (i.e. Pebble).

6.3 Countermeasures of Health Vulnerabilities

In this section we provide guidelines to avoid each of these health vulnerabilities, by means of literature support from previous studies in the field and comparison (if available) with a healthy ecosystem for validation.

6.3.1 Partner Mobilization

Our findings put the emphasis on the importance of having consistent and lasting partnerships. As Hedman and Kalling (2003) declared, external actors can potentially be partners or competitors, the line between both business dynamics is thin. The data briefly shows the concept of cross-platform collaborations that might have helped Blackberry and Windows phone survive longer than expected since it helped them add a portion of Android users to their customer base which they wouldn't have without the collaboration.

Waltl, Henkel and Baldwin (2012) indicate in their research that intra-platform approach increases the attractiveness of the ecosystem and thus increase the end user base. One of their interviewees mentioned that making it easier for partner ecosystems to develop add-on and extension made their ecosystem larger. This statement further adds to our own data, with the customer view we have in our analysis we concluded the importance of making add-on and extensions easy to implement for partners (e.g. in the Firefox and Yahoo case study). Additionally, ecosystem partners know their source code better and should be able to give a better implementation of their product to integrate to the candidate ecosystem. As an example to this statement, the Facebook application on Windows phones was created by Microsoft instead of Facebook itself. Hence, all updates were not available and we assume that the source code was not the same as in the original application. The Application always was at a beta version and did not have full feature and users cannot keep up with the same advances that Android or iOS phone users have. However, the situation changed when the official version was finally integrated in the middle of the year 2016.

With this research we came to the conclusion that a dynamic and and friendly partnership with both competitors and potential partners is a defining step in the evolution of an ecosystem. Most end users have accounts in multiple platforms and nowadays centrality is a concept in high demand. We want everything to be accessible in one click and to be able to connect to each account without issue of compatibility or partnership restrictions due to competitiveness. As Bengtsson and Kock (2000) define in their paper "coopetition" as competitors both compete and cooperate with each other. Coopetition is needed in the ecosystems in order to combine pressure to develop within new areas provided by competition and access to resources provided by cooperation (Bengtsson & Kock, 2000).

As contrast on healthy ecosystems practice in terms of partnerships, Gueguen, Pellegrin-Boucher and Torres (2006) provide a good example on how SAP managed their alliances with rival companies to keep their ecosystem healthy. Although their research was conducted 12 years ago, the observation and conclusion made for the SAP case were still applicable years later (van Angeren, Kabbedijk, Jansen & Popp, 2011). SAP secured it's own position in the market and established a healthy and strong ecosystem by conforming to all four collective strategies distinguished by Astley and Fombrun (1983).

The two most relevant collective focusing on partnerships are *the agglomerate strategies* which is the indirect association between rival companies and *the confederate strategies* which represents rival companies that develop partnership agreements. *"In terms of agglomerate strategies: From the end of the 1980s SAP's solutions were offered to most of the largest consultancy firms. These companies, however, are SAP's rivals both in terms of the services offered to companies and because they offer competitive software packages to SAP (Oracle, People Soft, JD Edwards, Baan).*

For confederated strategies: SAP made an alliance with IBM (its rival in the field of computer engineering) by jointly developing a new, integrated e-business solution." (Gueguen et al., 2006).

6.3.2 Platform Security

With the platform security we are looking at two aspects. First, the understanding of the user in terms of security and privacy and how it is provided and presented to them. Second, what are organizations and platforms doing to insure the security of their users data and fight malicious activities targeting them. The common subject encountered with our case studies is the ease of hack faced by users and organizations. Understandably, an ecosystem which is not secure, is a vulnerable ecosystem which can easily be harmed resulting in it becoming unhealthy and inevitably loosing actors (i.e. partners, developers, and customer).

Payne raised the awareness about "back door" hacking. He defined it as a "malicious code either inserted into, or in some way attached, to a legitimate program or system that allows an attacker to easily and covertly bypass existing security mechanisms" (Payne, 2002, p. 64). However, he makes two interesting observations. First, certain countries have "distrust" over the use of software produced by other countries, and second, open source code makes it impossible to hide the malicious bit of code, since everyone can check the code for themselves. But the understanding of the malicious code, the ways of identifying vulnerabilities and its effect on the platform seems to be challenging (Bosch, 2010).

Security proves to be the most challenging in term of finding a validation among healthy ecosystems. A previous study has shown that no matter the level of health of your ecosystem, the practice of security checks for applications in the App store world is different from one case to another (Jansen & Bloemendal, 2013). We consider both Google and Apple to be healthy ecosystems. However, out of the two, only the Google App store performs app security integration and app security report.

Furthermore, it seems that hackers are getting better at performing breaches with the evolution of technology. A set of steps to follow is provided by security experts to avoid a breach and/or to deal with the aftermath of a breach (Lord, 2018). At last, security expert say that establishing a good communication with all actors of the ecosystem is the most important first step to take.

We therefore, conclude that to avoid this health vulnerability, the main organism of the ecosystem should be responsible of making its platform as secure and safe as possible. To that effect, **a)** investing in good security system is required, additionally, **b)** the entry barrier should include a security check of the code (i.e. mobile application, add on, extensions). And finally, **c)** more seminars and conferences about security should be organized to raise the interest of all actors around the subject and to keep innovating in terms of protective system by continuously challenging it and pushing its limits.

6.3.3 Product Quality

As Jansen (2014) found out in previous research, the fastness of bug fix timing and the ability for customer to easily report is a sign of a project health which is not to be confused with ecosystem health, but we consider with our current research that it does influence the ecosystem health to some extent.

Furthermore we determine user testing as an important step of the production phase. If skipped, it can be costly with extra fixes, repairs and returns after the users declare (multiple) issues. Many large sites run a multitude of experiments each year testing all aspects of a software or product put in commercial use by the platform (Kohavi & Longbotham, 2017). According to the budget of the organization, the testing can be a heuristic evaluation which is a specialist reports method, user testing which is an observational method, or a combination of both. Fu, Salvendy and Turley (2002) found out that "user testing is more effective in discovering usability problems that novice users encounter". However, they mentioned that a combination is best to rule out all problems, but the choice is also depending on the budget of the company and the goal of the design phase in term of usability.

Our research data pointed out that some of the cases we studied do provide assistance to developers with compatibility, usability or functional testing of their applications with an online controller. The use of online controlled experiments is getting more popular among startups and smaller companies becoming a critical tool in defining change, but it appears to be important in large organizations as well (Kohavi et al., 2009).

We conclude that this vulnerabilities is mainly organizational and can be resolved by **a)** conducting research in order to find out what customers find important resulting in the creation of customer-specific products, **b)** enlarge the organization with more competent employees and invest in perfecting their skills to improve the internal quality of the organization, **c)** improve the response time of bug fixing, and **d)** provide assistance to developers with compatibility, usability or functional testing of their applications before publication.

6.3.4 Customer Care

The information unraveled from our data is that users do not find it easy to understand the use of certain products or features. Certain subjects faced by users should not be impossible to locate on a manual page. However, users turn to forums and to the "knowledge" of other users in the forum to help them fix their problems. Therefore, we find it helpful and essential to have a search system in the forum in order for users to locate the discussion or the tutorial they need in a fast and reliable way.

Furthermore, having a fast responsive team for customer care would be a positive change in order for the end user to feel respected instead of being neglected or unimportant. Additionally, in light of a big issue that touches the ecosystem and the user's data (i.e. a hacked platform), a team should immediately be immobilized to give full attention to the customer concerns.

As an example, Apple inc faced in 2009 the consequences of not giving attention to customers. With the release of iPhone 4, the external antenna gate was not well received by the public, and customers kept on complaining about signal issues. Ira Kalb, clinical marketing professor at the USC Marshall School of Business, declares that Apples is one of the few companies that have a well-regarded approach to customers realizing it's importance as return on investment item. In an interview article (Ogg, 2010), he discusses the event and points out the good an bad points on the story.

- Steve job's initial response was to blame the user for not using the phone correctly and indicating that holding the phone in a different way would not introduce the issues faced by the customers.

- Then the blame was directed towards a software issue that would be fixed by an upcoming update.

However, the unsatisfied customers resulted in a published official bad review of the mobile. At that point, Apple cannot deny the risk of harming it's customer satisfaction strength, and held a press conference (Lowensohn, 2010) and offered free bumper cases that supposedly would solve the issue, and promise a refund if asked for. "The lesson was a useful one for a company that already handles its customer interactions mostly well", said Kalb. And the company continues to improve it's customer interaction and satisfaction since then.

6.3.5 Market Awareness and Competitiveness

It is common knowledge that every organizations has competitors. The fact of being the first to release a technology never seen before does not guarantee the monopoly on the market. Competitors can improve the technology and create a wider customer base. Therefore, an ecosystem should be aware of its competitors and be able to keep the race going and keep a competitive advantage of always innovating.

Porter and Porter (1979) introduced the threat of entry, defining it seriousness depending on the preexisting barriers faced by the organization and the reaction to expect from existing competitors. Furthermore, organizations should keep innovative strategies in place in order to give a competitive product that keeps improving in terms of quality and cost (Hedman & Kalling, 2003).

Finally, we consider market awareness and innovation to be an important step to keep a healthy business ecosystem and contribute to the elimination of our last health vulnerability.

6.4 Implications

To conclude this chapter, we consider that all our cases have made terrible decisions that put their competitors in the lead instead of them and eventually drive them to their deaths.

- Google and Apple are dominating the phone market compared to Microsoft and Blackberry. Microsoft's downfall in trying to revives its smartphone line continued when they achieved a worst outcome than expected by acquiring Nokia and selling it for cheaper three years later. However our cases study included only the mobile phone part of the ecosystem, the rest of the ecosystem in terms of software product, system solution and operating system, does not seem to have this level of issue. This drives us to conclude that Microsoft should focus on what it is good at, and stop trying to revive a dead part of it's ecosystem that might harm the rest of it. Additionally, Blackberry can simply not compete with the new technologies provided by the others (A.12).

- Google strikes again by providing a better and worldwide used browser, in comparison to Firefox with a 6 times higher percentage of the market share (A.11).

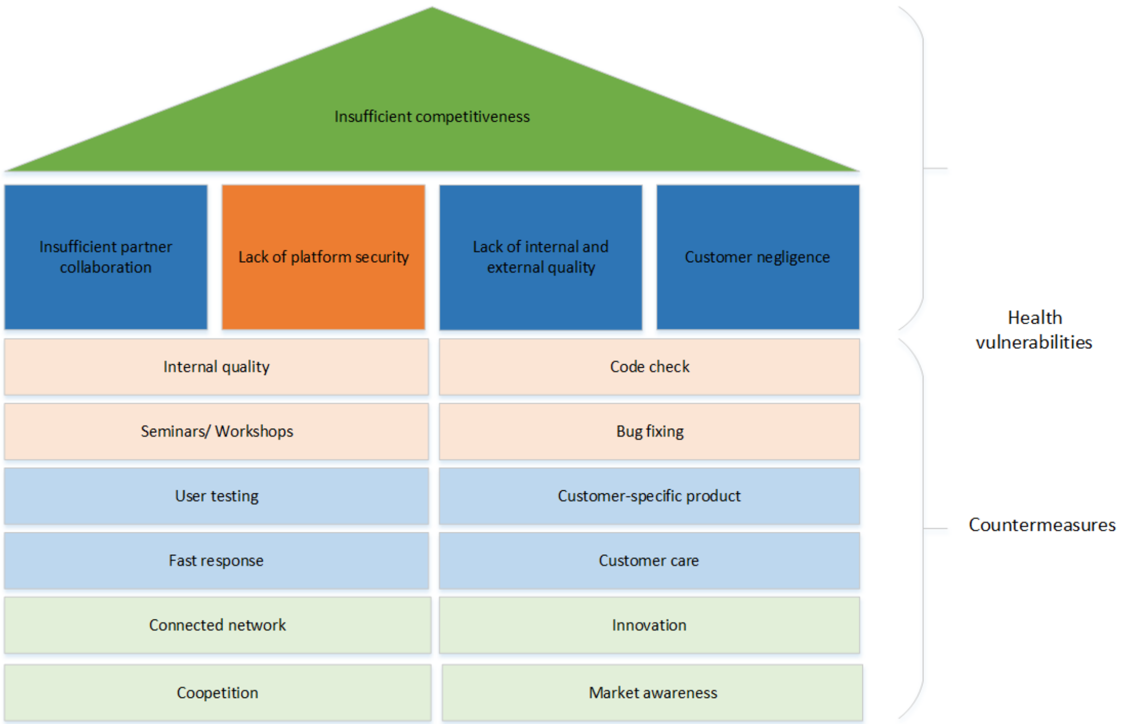
- When it comes to the smartwatch market, the market is overflowing with different providers and different ranges of prices that equally compete among each other with quality price ratio. Considering the price of a pebble the direct competitors is Apple, controlling the market share closely followed by Samsung (Stark, 2017).

- Yahoo's main issue was to be too wide spread without knowing were to shift their focus and were to perfect its technology and innovation, as Kim (2016) points out. Making it facing too many competitors in a wide range of fields including news provider, email provider, search engine, e-commerce, social media, analytics, and the list goes on. At last, Yahoo is now under a restructuring plan that will narrow its focus to three platforms (search, email, Tumblr) and four content verticals (news, finance, sports, and lifestyle), as well as its Gemini and Brightroll ad offerings. (Kim, 2016)

We finally, present our diagram as a list of identified health vulnerabilities and guidelines on how to avoid them, showing their interconnection and dependencies to each other (figure 6.3). This model can be interpreted with a bottom-up approach . It displays **a)** the missing practices (countermeasures) that would lead to **b)** the health vulnerabilities.

The health vulnerabilities represent the pillars. The ones highlighted in blue indicate the conclusions made from the customer perspective view. The green represents the conclusions made from the market and business study. The orange colour is for the conclusions derived from the combination of both the market and the users.

The guidelines are the steps that have to be implemented to avoid the health vulnerabilities. Similarly, the guidelines represent a change made at a customer level in light blue, ecosystem level (e.g business strategies, actor roles, events) in light green and internal level in light orange.



Health vulnerabilities
- Business perspective
- Customer perspective
- Customer and business perspective

Countermeasures
- Ecosystem level change
- Customer level change
- Internal level change

Figure 6.3: Bottom up visualization of our findings and results

Chapter 7

Discussion

7.1 Other Findings

Along with the qualitative content analysis of the forums, we have analyzed the developer network which provided positive results shared by all cases. We included the study of the connectivity of the networks since it is an important metric in the robustness pillar. We looked at the opportunities provided by the platforms to allow its actors, the developers, to interact. These opportunities include facilitating communication by creating a developer forum, creating a Github profile where developer can collaborate on various projects, and finally by being part of the growing community.

All five case studies have forums and portals for developers where support, developing tools, blogs, and tutorials are available. Additionally they all have a Github profile for open source collaborations. The figures from A.7 to A.10 in Appendix A represent the network connections in terms of following and followers of collaborators on the Github repositories of four out of 5 cases. Unfortunately Windows phone Github social graph could not be generated due to a high number of accounts available on the repository which saturated our query.

Microsoft introduced CodePlex [1] as part of the developer community cooperation in 2006 and migrated it to Github by 2017 because it became more prominent and widely used by developers. The CodePlex highly resembles Github in term of forking startgazing and collaborating. Currently, the webpage is on read only but users can find all previous and new repositories in the Github account of Microsoft [2]. Although there is no clear distinctions between repositories dedicated to windows phone and repositories for all other products of Microsoft, developers can filter projects per type of programming language through the search bar. The Microsfot Github profile counts 1,831 repositories and 3824 people active on them.

Similarly Firefox does not have it's own Github profile but is part of the Mozilla Github[3], where there is no distinctions made by the platform, but repositories are easily found via the search bar. The Mozilla Github counts 1797 repositories and 292 collaborating people.

Comparably, Yahoo and Blackberry have a fairly smaller amount of repositories and active users, respectively 266 and 147 repositories with 27 and 86 people collaborating on them. Out of the 86 people in Blackberry's Github circle, only 14 form a unidirectional or bidirectional connection, the rest appears to be individual spots in the ecosystem not interacting with their peers.

Finally, the Pebble Github seems to be the least popular with only one active collaborator with 131 repositories. Although, another more popular Github profile[4] exists with 13 active people collaborating on 22 repositories. The above mentioned Github profiles are the one provided and created by the platforms themselves. However it is possible that not all freelance developers are

[1] https://opensource.microsoft.com/
[2] https://github.com/Microsoft
[3] https://github.com/mozilla
[4] https://github.com/pebble-dev

collaborating on them. As an example, the pebble Github shows only one moderator, but certain repositories have multiple stargazers or forks created by other collaborators.

Open source software companies rely in big part on third party developers to grow their ecosystem size. Those developers are mainly freelancers implementing applications for public use by publishing them in the appropriate online store of the chosen platform. However, all stores have an entry barrier in form of a check list to make sure that the developed component is of high quality and therefore won't jeopardize the stability of the ecosystem and it's health. As mentioned by Van Den Berk et al. (2010) "Choosing the right balance between quality and innovation is vital for ecosystem health".

Network connectivity is mostly discussed in previous studies with open source companies. However, all our cases are not fully open source companies, but harbour certain open source projects following the OSS movement that all big organizations initiated when the popularity of the community grew. OSS developers collaborate with each other from around the world and prefer to be part of a successful project (Madey, Freeh & Tynan, 2002). This aspect shows that the predetermined health of an ecosystem is important for the developers to choose to join it. Additionally, it can influence the *multiple market* health metric of the niche creation pillar since the variety of collaborators brings the diversity of languages and cultures.

Our data shows that all our studied platforms have made sure to build a developer network by providing portals including tutorials, download and installation assistance, discussion forums, and an official Github profile with projects to collaborate on. As mentioned previously, events are important to keep the developers engaged, and gain new knowledge. Apart from Pebble, all our studied platforms organize meetings or events to gather third party developers to discuss important subjects and release of new products. However, our current conclusion on the developer's network is incomplete. Due to time constraints, we cannot validate our findings by comparing it to a healthy ecosystem (e.g. SAP, Apple or Google).

7.2 Limitations and Threats to Validity

Although the research was conducted by means of established research methods, it is necessary to assess its quality in order to enable transparent interpretation of the results. In this section, we discuss the limitations we have encountered during the research in addition to the reliability, construct, internal and external validity defined by Yin (2009)

The first limitation met during this research was the use of python as a mining language for forums and the incapacity of mining dynamic websites. Choices have been made to insure the continuity of the project and limit the loss of time:

- We turned to another programming language (c#) when it came to the forums. This decision makes the methodology more reliable since the data collection can be replicated with different tools.
- We had to eliminate dynamic websites such as Stack-overflow from our list of data sources. This decision brought a limitation in term of the study of the developers and restricted our analysis to mainly customer's views and experiences.

The second limitation concerns the data set since the 5 cases had different time lines compared to each other, Pebble had available data from 2013 to 2016 only, whereas the study of Firefox provided data from 2006 to 2018. This resulted in having greater number of data entries in some cases than others, and we theorize that the conclusion might differ if other cases are added to the research.

Construct Validity: is involved with operational measures, and whether they are suitable to measure theoretical constructs (Yin, 2009). In this research, our health vulnerabilities are deducted from nodes that were subject of analyzed data. The selection of nodes is decided upon relying on knowledge obtained from previously conducted studies. Additionally, a chain of evidence to our results can be trailed to the document database, the observations made, and finally the sources used to extract it from. Furthermore to increase the accuracy and completeness of our data, we triangulate our collection method by combining multiple sources such as social media, web forums, grey literature, market share releases, and press releases. A limitation of our study is that we do not include the exact creation time of each subjects. Instead of days and hours, we are grouping data in the same months together and relate it to real time events happening around that period.

Internal Validity: regards casual relationships, if applicable, are properly linked and omit the influence of other variables (Yin, 2009). Our data is collected from public sources, thus we assume that participants are not influencing our measurements since they are not aware of their published posts and online discussions being studied. As mentioned earlier the limitation in this prospect is that certain sources used for collection did not authorize the crawling technique. That represents a number of subjects that could not be included in our analysis.

External Validity: concerns the generalization of the finding (Yin, 2009). A multiple case study design was chosen to benefit generalization. The exclusion and the inclusion of cases have been met by providing a set of criteria derived from literature research. Furthermore, our chosen cases represent different types of ecosystems such as mobile ecosystem and web provider ecosystem.

Reliability: of case study research is involved with replicability of a research (Yin, 2009). The steps of our research approach and the methodology is thoroughly described in the appropriate chapters. A database has been set to store all relevant case study data from raw data to interpreted data. Accordingly, all findings have been cross-referenced to increase transparency and traceability.

In addition to the limitations and validity, general critiques about the quality of this research should be addressed. Due to time constraints, we have not validated our results with experts but rather with papers and general public data regarding a couple of healthy ecosystems. Therefore, we consider the validation of the list of health vulnerabilities using expert interviews or using more focused case study would be an added value to this research. Furthermore, we discuss in the next section what we believe should be done in the future to make this research bigger and more complete.

7.3 Future Work

Relatively much time was spent developing a suitable tool to satisfy our data collection and analysis. Future research (possibly using the same tool or a similar procedure) could be more in-depth, focusing on the following areas:

- Data collection: since the crawling of dynamic websites was not possible in the present research, including those websites in future research would give an even greater data set to analyze.
- Developer network: a description of the developer ecosystem health could be included. Additionally, data could be collected from multiple developers hubs and the analysis of personal comments and experiences of developers regarding dead ecosystems. Finally, a comparison with a healthy ecosystem could give a bigger picture of the findings.
- Validation: the validation of the platform security is missing the element of comparison to a healthy ecosystem. Furthermore, expert opinion could give a broader insight.
- Follow up: considering that the last data was collected by may 2018, changes in some of our case were noticed following that time period. Microsoft has acquired Github [5], and Yahoo decided to narrow its focus to three platforms. Do those major changes impact our current findings?

[5] https://news.microsoft.com/2018/06/04/microsoft-to-acquire-github-for-7-5-billion/

- Scope: Our findings focus on public data obtained mainly from the users with a smaller amount of data from the developers. Future work including interviews with the companies major stakeholders might give contrasting findings.

Chapter 8

Conclusion

We extracted information about ecosystem health evolution that is implicitly available in publicly accessible data to answer our main research question : How can software ecosystem health vulnerabilities be identified and eliminated?

In order to do so we started by answer our three sub-questions first.

Our first sub-question *Q1: what are the characteristics of healthy and/or unhealthy ecosystems?*, was answered by extracting health metrics from literature review. The literature study included study and comparison of business ecosystem, software ecosystem and open source ecosystem health frameworks in order to determine the similarities between them. In this section of the research, we found out that frameworks share a number of similar health metrics that have been used to identify our cases.

Our next sub-question *Q2: which unhealthy ecosystems can be studied and how to choose them?* This question was answered by applying the detected health metrics from the first sub-question to our list of 16 preliminary cases. Each case was scanned by applying case selection criteria, and 5 cases where finally selected for the remainder of the research.

The third sub-question *Q3: what are observable weaknesses in software ecosystems?* was answered with the help of qualitative analysis of the data mined through various sources and by studying data available online acquired using data mining techniques including Python and C# scripts.

We have identified five health vulnerabilities, and each had multiple counter measures derived from an extensive data analysis of unhealthy cases and literature review. First, insufficient partner collaboration which was identified by determining all nodes that discussed weak or nonexistent partnerships. This included the absence of partnership between a similar company or platform and one of our unhealthy case study, and the incompatibility of features and application provided by rival companies. This health vulnerability can be avoided by adopting coopetition, a notion introduced by Bengtsson and Kock (2000), or by make smart business decision including agglomerate strategies and confederate strategies. However, confederate strategy should be considered using a company which is well established as well (i.e. in our validation case, the SAP alliance with IBM) and not a company in a bad position that can jeopardize your position even more and lose money by reselling it later after a failed attempt to revive a line of product (Hern, 2016) (i.e. the alliance between Microsoft and Nokia).

Secondly, we identify lack of platform security. This health vulnerability was identified mainly by the collected data pointing at the big security breach of the 21st century faced by our case study Yahoo (Armerding, 2018). Our finding show that even with Yahoo providing security tips to their users [48], the change still had to be made internally by implementing better security systems. Furthermore this applies also to the other cases, by providing a security check in addition to the entry barrier of third-party developers joining the ecosystem. Finally, we have discovered a practice done by Blackberry including workshops and hackatons (i.e. hacking challenges to test the security system put in place), which in our opinion, is a good initiative to keep up to date with the hacking technologies and to provide stakeholders and end users knowledge concerning their security.

Our third health vulnerabilities, touches the subject of product quality. With this health vulnerability we learned that our cases should have a better and higher entry barrier in terms of competences. Updates and bug fixes should be done to not only meet the deadlines posed by the organization but also to meet the expectation of the end user in receiving a fully functioning end product. Additionally, research on customer expectation should be conducted to have a better customer response to features and functionality. This relates us to the firth health vulnerability, the customer negligence. In the software ecosystem field, the end user is not considered as an actor most of the time. Our research showed that this negligence of the customer and prioritization of other stakeholders in ecosystems is indeed what harms its health. Companies often think about the technology and not the end user, excluding their needs and feedback from all process (i.e. user testing, delivery, user satisfaction). Furthermore, lessons can be learned from companies that do understand the importance of a customer's need and satisfaction.

The last health vulnerability, sums all four vulnerabilities and relates to the whole ecosystem and the dynamic with others regarding ecosystem governance. The lack of market awareness and competitiveness is what strikes the most in our analysis. Although, our chosen case studies have been in the market before their competitors, it is those competitors who dominate the market in their consecutive fields.

In conclusion, this research is to our knowledge the first of its kind to use the ecosystem's actors view to identify the threats and vulnerabilities to its health. It contributes to a better understanding of maintaining that healthy status of the ecosystem by highlighting aspects that are not usually included in ecosystem health frameworks. The output of this paper has an added value to the research community since we provide a method to objectively identify the health vulnerabilities. We studied the ecosystem of five cases and analyzed data of over 650,000 posts and tweets published in the period between January 2006 and February 2008 using natural language processing. Finally, we believe our analysis would increase the awareness about the position and influence of end users in an ecosystem.

Notes

[1]https://yahoo.tumblr.com/post/54125001066/keeping-our-focus-on-whats-next
[2]https://advertising.aol.com/platforms
[3]https://techcrunch.com/2016/01/25/one-by-aol-publishers/
[4]https://www.oath.com/our-brands/
[5]https://www.onebyaol.com/about
[6]https://www.statista.com/statistics/263439/global-market-share-held-by-rim-smartphones/
[7]http://developers.deezer.com/guidelines/countries
[8]https://www.prnewswire.com/news-releases/ebay-inc-reports-third-quarter-2017-results-300539211.html
[9]https://www.eclipse.org/eclipse/news/4.7/
[10]http://www.eclipse.org/forums/
[11]http://events.eclipse.org/
[12]http://marketplace.eclipse.org/
[13]https://andreasgal.com/2017/07/19/firefox-marketshare-revisited/
[14]https://developers.google.com/picasa-web/
[15]http://www.spiremarketresearch.com/global-smartglasses-market-2017-demand-insights-key-palyers-segmentation-and-forecast-to-2022/#request_for_sample
[16]https://variety.com/2016/digital/news/time-inc-myspace-viant-1201703860/
[17]https://blog.getpebble.com/2016/12/07/fitbit/
[18]https://techcrunch.com/2013/01/24/nokia-confirms-the-pure-view-was-officially-the-last-symbian-phone/
[19]https://www.nokia.com/sites/default/files/files/q4-2012-results-presentation-pdf.pdf
[20]https://www.visualstudio.com/en-us/news/releasenotes/vs2017-relnotes
[21]https://blogs.msdn.microsoft.com/developer-tools/
[22]https://www.visualstudio.com/vs/compare/
[23]https://www.cnet.com/news/windows-10-mobile-features-hardware-death-sentence-microsoft/
[24]https://www.statista.com/statistics/216573/worldwide-market-share-of-search-engines/
[25]https://www.statista.com/statistics/267161/market-share-of-search-engines-in-the-united-states/
[26]https://globalnews.ca/news/860689/blackberry-timeline-a-look-back-at-the-tech-companys-history/
[27]http://www.bbc.com/news/technology-37493566
[28]https://web.archive.org/web/20100513062601/http://www-archive.mozilla.org/MPL/relicensing-faq.html
[29]https://www.cnet.com/news/ibm-names-firefox-its-default-browser/
[30]https://www.mozilla.org/en-US/foundation/licensing/
[31]https://techcrunch.com/2016/12/07/pebble-confirms-its-shutting-down-devs-and-software-going-to-fitbit/
[32]https://mobiforge.com/timeline/windows-phone-history
[33]https://news.microsoft.com/2011/02/10/nokia-and-microsoft-announce-plans-for-a-broad-strategic-partnership-to-build-a-new-global-mobile-ecosystem/
[34]https://www.forbes.com/sites/timworstall/2013/09/08/the-real-reason-microsoft-bought-nokia-transaction-costs/#21d721a83360
[35]https://www.theverge.com/2014/2/24/5440498/nokia-x-android-phone-hands-on
[36]https://www.cnet.com/news/windows-10-mobile-features-hardware-death-sentence-microsoft/
[37]http://www.zeebiz.com/companies/news-how-did-yahoo-fall-from-125-billion-to-5-billion-in-15-years-4200
[38]https://www.securityfocus.com/news/254
[39]https://www.pcworld.com/article/116858/article.html
[40]https://gigaom.com/2005/06/14/breaking-yahoo-buys-dial-pad/
[41]https://www.geek.com/news/verizon-and-yahoo-hook-up-to-offer-dsl-558651/
[42]https://searchengineland.com/optimizing-yourself-for-the-yahoo-microsoft-search-alliance-46800
[43]https://www.csmonitor.com/Technology/Horizons/2012/0712/Yahoo-hack-steals-400-000-passwords.-Is-yours-on-the-list
[44]https://www.nytimes.com/2016/09/23/technology/yahoo-hackers.html
[45]https://www.verizon.com/about/news/verizon-completes-yahoo-acquisition-creating-diverse-house-50-brands-under-new-oath-subsidiary
[46]https://www.zscaler.com/company/about-zscaler
[47]https://support.mozilla.org/en-US/questions/1068675
[48]https://safety.yahoo.com/index.htm

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30–38). 28

Armerding, T. (2018). *The 17 biggest data breaches of the 21st century.* Retrieved 2018-07-13, from https://www.csoonline.com/article/2130877/data-breach/the-biggest-data-breaches-of-the-21st-century.html 47

Astley, W. G. & Fombrun, C. J. (1983). Collective strategy: Social ecology of organizational environments. *Academy of management review*, *8*(4), 576–587. 37

Barbosa, O. & Alves, C. (2011). *A systematic mapping study on software ecosystems.* Citeseer. 3

Bengtsson, M. & Kock, S. (2000). " coopetition" in business networks—to cooperate and compete simultaneously. *Industrial marketing management*, *29*(5), 411–426. 37, 47

Bosch, J. (2010). Architecture challenges for software ecosystems. In *Proceedings of the fourth european conference on software architecture: Companion volume* (pp. 93–95). 38

Bosch, J. & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems. *Journal of Systems and Software*, *83*(1), 67–76. 1, 9

Ceron, A., Curini, L., Iacus, S. M. & Porro, G. (2014). Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, *16*(2), 340–358. 27

Costanza, R. (1992). Toward an operational definition of ecosystem health. *Ecosystem health: New goals for environmental management*, 239–256. 1, 9

Crowston, K., Howison, J. & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, *11*(2), 123–148. 14

Deng, S., Sinha, A. P. & Zhao, H. (2017). Adapting sentiment lexicons to domain-specific social media texts. *Decision Support Systems*, *94*, 65–76. 28

den Hartigh, E., Tol, M. & Visscher, W. (2006). 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). 12, 14, 15, 16, 17, 28

Fa, J. E., Olivero, J., Farfán, M. Á., Márquez, A. L., Vargas, J. M., Real, R. & Nasi, R. (2014). Integrating sustainable hunting in biodiversity protection in central africa: hot spots, weak spots, and strong spots. *PLoS One*, *9*(11), e112367. 1

Franco-Bedoya, O., Ameller, D., Costal, D. & Franch, X. (2014). Queso a quality model for open source software ecosystems. In *Software engineering and applications (icsoft-ea), 2014 9th international conference on* (pp. 209–221). 10, 14, 15, 16, 17, 28, 33

Franco-Bedoya, O., Ameller, D., Costal, D. & Franch, X. (2017). Open source software ecosystems: A systematic mapping. *Information and software technology*, *91*, 160–185. 3

Fu, L., Salvendy, G. & Turley, L. (2002). Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, *21*(2), 137–143. 38

Garousi, V., Felderer, M. & Mäntylä, M. V. (2016). The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In *Proceedings of the 20th international conference on evaluation and assessment in software engineering* (p. 26). 19

Goeminne, M. & Mens, T. (2013). Analyzing ecosystems for open source software developer communities. *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*, 247–275. 14

Gueguen, G., Pellegrin-Boucher, E. & Torres, O. (2006). Between cooperation and competition: the benefits of collective strategies within business ecosystems. the example of the software industry. In *Eiasm 2nd workshop on co-opetition strategy, milan, italy* (pp. 14–15). 37, 38

Hedman, J. & Kalling, T. (2003). The business model concept: theoretical underpinnings and empirical illustrations. *European journal of information systems*, *12*(1), 49–59. 37, 39

Help-nv11.qsrinternational.com. (2018 (accessed April 11, 2018)). How does pattern-based coding work? [Computer software manual]. Retrieved from http://help-nv11.qsrinternational.com/desktop/deep_concepts/how_does_pattern-based_coding_work.htm 7

Help-nv11.qsrinternational.com. (2018 (accessed April 4, 2018)). Nvivo 11 for windows help - about nodes [Computer software manual]. Retrieved from http://help-nv11.qsrinternational.com/desktop/concepts/about_nodes.htm 7

Hern, A. (2016, May). *Nokia returns to the phone market as microsoft sells brand.* Retrieved 2018-07-13, from https://www.theguardian.com/technology/2016/may/18/nokia-returns-phone-market-microsoft-sells-brand-hmd-foxconn 47

Hoving, R., Slot, G. & Jansen, S. (2013). Python: Characteristics identification of a free open source software ecosystem. In *Digital ecosystems and technologies (dest), 2013 7th ieee international conference on* (pp. 13–18). 14

Hyrynsalmi, S., Mäkilä, T., Järvi, A., Suominen, A., Seppänen, M. & Knuutila, T. (2012). App store, marketplace, play! an analysis of multi-homing in mobile software ecosystems. 2

Hyrynsalmi, S., Suominen, A. & Mäntymäki, M. (2016). The influence of developer multi-homing on competition between software ecosystems. *Journal of Systems and Software*, *111*, 119–127. 2

Iansiti, M. & Levien, R. (2004). Strategy as ecology. *Harvard business review*, *82*(3), 68–81. 1, 10, 12, 14, 28

Jansen, S. (2014). Measuring the health of open source software ecosystems: Beyond the scope of project health. *Information and Software Technology*, *56*(11), 1508–1519. 10, 14, 15, 16, 17, 28, 33, 38

Jansen, S. & Bloemendal, E. (2013). Defining app stores: The role of curated marketplaces in software ecosystems. In *International conference of software business* (pp. 195–206). 38

Jansen, S., Finkelstein, A. & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. In *Software engineering-companion volume, 2009. icse-companion 2009. 31st international conference on* (pp. 187–190). 1, 9

Joblin, M., Apel, S., Hunsen, C. & Mauerer, W. (2017). Classifying developers into core and peripheral: An empirical study on count and network metrics. In *Proceedings of the 39th international conference on software engineering* (pp. 164–174). 28

Kabbedijk, J. & Jansen, S. (2011). Steering insight: An exploration of the ruby software ecosystem. *Software Business*, 44–55. 2

Kim, E. (2016). *Yahoo's list breaking down its competition shows the company still has no idea what it really is.* Retrieved 2018-07-14, from http://www.businessinsider.com/yahoo-still-has-no-idea-what-it-is-2016-8?international=true&r=US&IR=T 40

Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J. L. & Melamed, T. (2009). Online experimentation at microsoft. *Data Mining Case Studies*, *11*. 39

Kohavi, R. & Longbotham, R. (2017). Online controlled experiments and a/b testing. In *Encyclopedia of machine learning and data mining* (pp. 922–929). Springer. 38

Koshland, D. E. (2002). The seven pillars of life. *Science*, *295*(5563), 2215–2216. 9

Kraft, R. & Myllymaki, J. P. (2009, April 14). *System and method for enhanced browser-based web crawling.* Google Patents. (US Patent 7,519,902) 27

Lord, N. (2018). *Data breach experts share the most important next step you should take after a data breach in 2014 - 2015 beyond.* Retrieved 2018-

07-11, from https://digitalguardian.com/blog/data-breach-experts-share-most-important-next-step-you-should-take-after-data-breach-2014-2015 38

Lowensohn, J. (2010). *Live blog: Apple gives free cases to iphone 4 owners.* Retrieved 2018-07-12, from https://www.cnet.com/news/live-blog-apple-gives-free-cases-to-iphone-4-owners/ 39

Lucassen, G., van Rooij, K. & Jansen, S. (2013). Ecosystem health of cloud paas providers. In *International conference of software business* (pp. 183–194). 2, 14

Lungu, M. F. (2009). *Reverse engineering software ecosystems* (Unpublished doctoral dissertation). Università della Svizzera italiana. 1, 9

Madey, G., Freeh, V. & Tynan, R. (2002). The open source software development phenomenon: An analysis based on social network theory. *AMCIS 2002 Proceedings*, 247. 44

Manikas, K. & Hansen, K. M. (2013a). Reviewing the health of software ecosystems–a conceptual framework proposal. In *Proceedings of the 5th international workshop on software ecosystems (iwseco)* (pp. 33–44). 1, 9, 14, 15, 16, 17, 28, 33

Manikas, K. & Hansen, K. M. (2013b). Software ecosystems–a systematic literature review. *Journal of Systems and Software*, *86*(5), 1294–1306. 1, 3, 9

McDowell, N. G. (2011). Mechanisms linking drought, hydraulics, carbon metabolism, and vegetation mortality. *Plant physiology*, *155*(3), 1051–1059. 9

Neuendorf, K. A. (2016). *The content analysis guidebook.* Sage. 7

Ogg, E. (2010). *Secrets of apple's customer success.* Retrieved 2018-07-12, from https://www.cnet.com/news/secrets-of-apples-customer-success/ 39

Onwuegbuzie, A. J., Leech, N. L. & Collins, K. M. (2012). Qualitative analysis techniques for the review of the literature. *The qualitative report*, *17*(28), 1. 5

Payne, C. (2002). On the security of open source software. *Information systems journal*, *12*(1), 61–78. 38

Pinto, J. K. & Mantel, S. J. (1990). The causes of project failure. *IEEE transactions on engineering management*, *37*(4), 269–276. 2

Pitta, D. A. & Fowler, D. (2005). Internet community forums: an untapped resource for consumer marketers. *Journal of Consumer Marketing*, *22*(5), 265–274. 27

Poon, P. & Wagner, C. (2001). Critical success factors revisited: success and failure cases of information systems for senior executives. *Decision support systems*, *30*(4), 393–418. 2

Porter, M. E. & Porter, M. E. (1979). How competitive forces shape strategy. 39

Radulovic, F. & Garcia-Castro, R. (2011). Extending software quality models-a sample in the domain of semantic technologies. In *Seke* (pp. 25–30). 14

Rapport, D. J., Costanza, R. & McMichael, A. (1998). Assessing ecosystem health. *Trends in ecology & evolution*, *13*(10), 397–402. 1

Russell, M. A. (2013). *Mining the social web: Data mining facebook, twitter, linkedin, google+, github, and more.* " O'Reilly Media, Inc.". 6, 28

Sarlan, A., Nadam, C. & Basri, S. (2014). Twitter sentiment analysis. In *Information technology and multimedia (icimu), 2014 international conference on* (pp. 212–216). 27

Soto, M. & Ciolkowski, M. (2009). The qualoss open source assessment model measuring the performance of open source communities. In *Empirical software engineering and measurement, 2009. esem 2009. 3rd international symposium on* (pp. 498–501). 14

Soussi, L., Spijkerman, Z. & Jansen, S. (2016). A case study of the health of an augmented reality software ecosystem: Vuforia. In *International conference of software business* (pp. 145–152). 2

Stark, C. (2017). *Wearables trendspotting: Apple is the smartwatch market.* Retrieved 2018-07-14, from http://www.starkinsider.com/2017/02/apple-smartwatch-market-q4-sales-report.html 40

Stieglitz, S. & Dang-Xuan, L. (2013). Emotions and information diffusion in social media—sentiment of microblogs and sharing behavior. *Journal of management information systems*, *29*(4), 217–248. 27

Tellis, W. M. (1997). Introduction to case study. *The Qualitative Report*, *3*(2), 1-14. Retrieved from http://nsuworks.nova.edu/tqr 5

Valença, G., Alves, C., Heimann, V., Jansen, S. & Brinkkemper, S. (2014). Competition and collaboration in requirements engineering: a case study of an emerging software ecosystem. In *Requirements engineering conference (re), 2014 ieee 22nd international* (pp. 384–393). 31

van Angeren, J., Kabbedijk, J., Jansen, S. & Popp, K. M. (2011). Partnership characteristics within large software ecosystems. *Proceedings of the Third International Workshop on Software Ecosystems*. 37

Van Den Berk, I., Jansen, S. & Luinenburg, L. (2010). Software ecosystems: a software ecosystem strategy assessment model. In *Proceedings of the fourth european conference on software architecture: Companion volume* (pp. 127–134). 2, 3, 36, 44

Van Lingen, S., Palomba, A. & Lucassen, G. (2013). On the software ecosystem health of open source content management systems. In *5th international workshop on software ecosystems (iwseco 2013)* (p. 38). 14

Waltl, J., Henkel, J. & Baldwin, C. Y. (2012). Ip modularity in software ecosystems: How sugarcrm's ip and business model shape its product architecture. In *International conference of software business* (pp. 94–106). 37

West, J. & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. *R&d Management*, *36*(3), 319–331. 2

Wnuk, K., Runeson, P., Lantz, M. & Weijden, O. (2014). Bridges and barriers to hardware-dependent software ecosystem participation–a case study. *Information and Software Technology*, *56*(11), 1493–1507. 2

Yin, R. K. (2009). *Case study research: Design and methods.* Sage publications. 5, 44, 45

Appendix A

Data charts

In this appendix, we display all the data charts referenced in the document, charts generated from data analysis, and Github social graph visualization of developer networks.

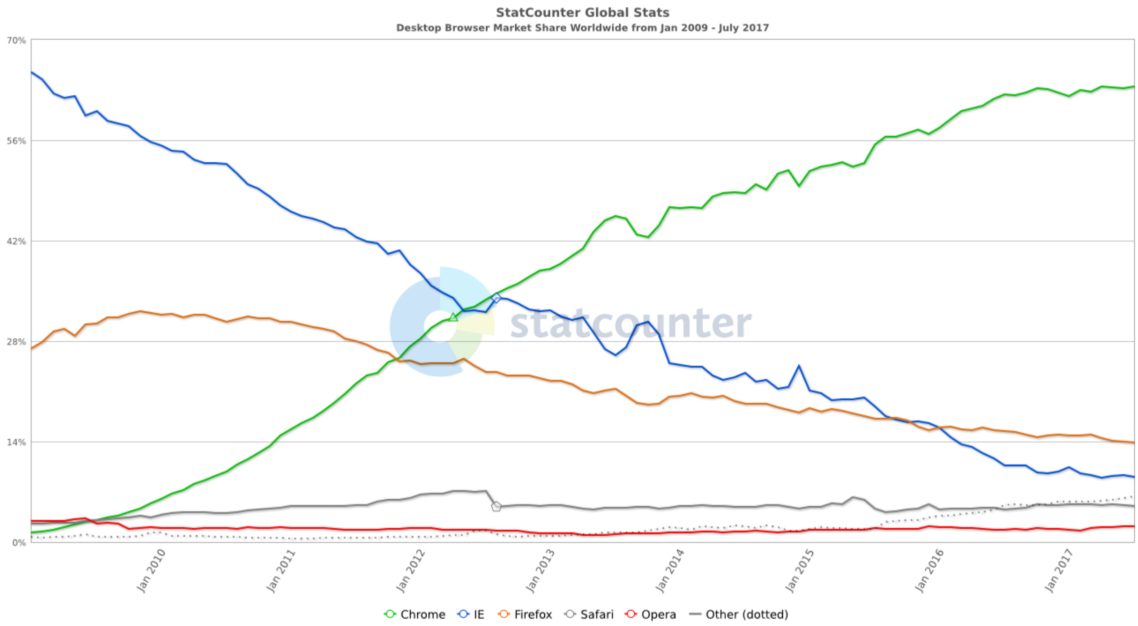


Figure A.1: Browsers comparison statistics from 2009 to 2017, retrieved from (StatCounter, 2018)

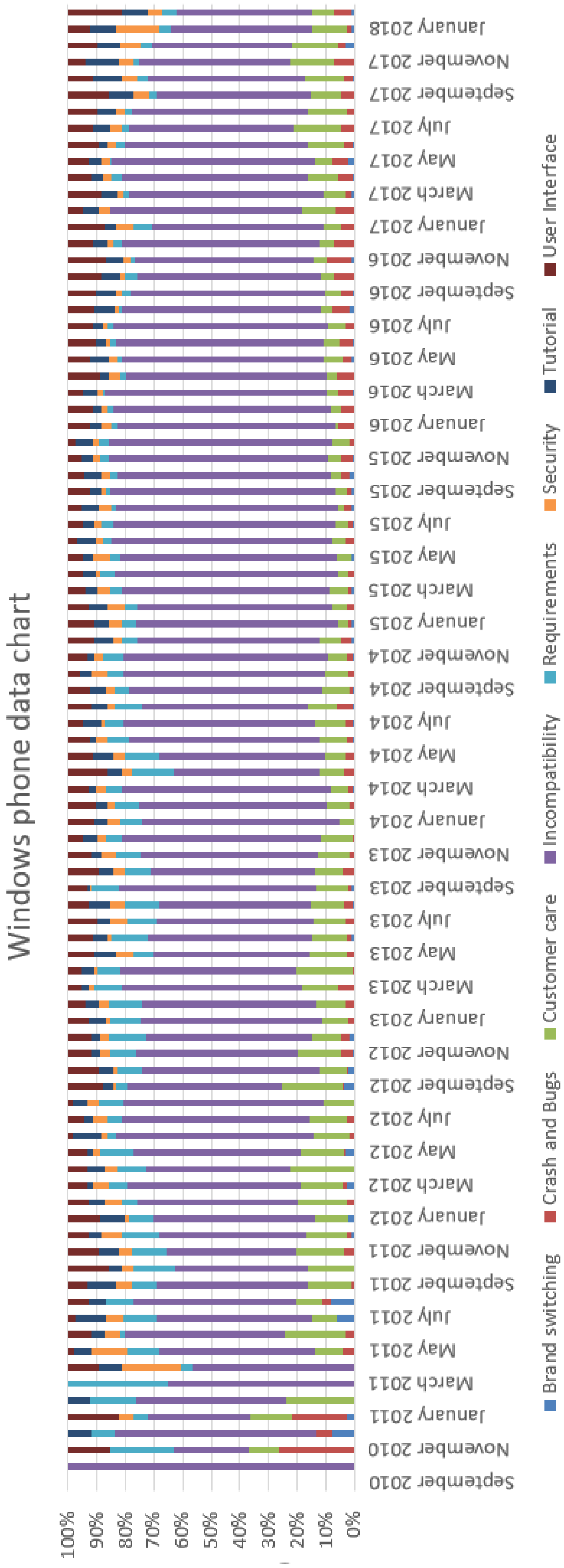


Figure A.2: Node's data chart of Windows Phone between 2010 and early 2018

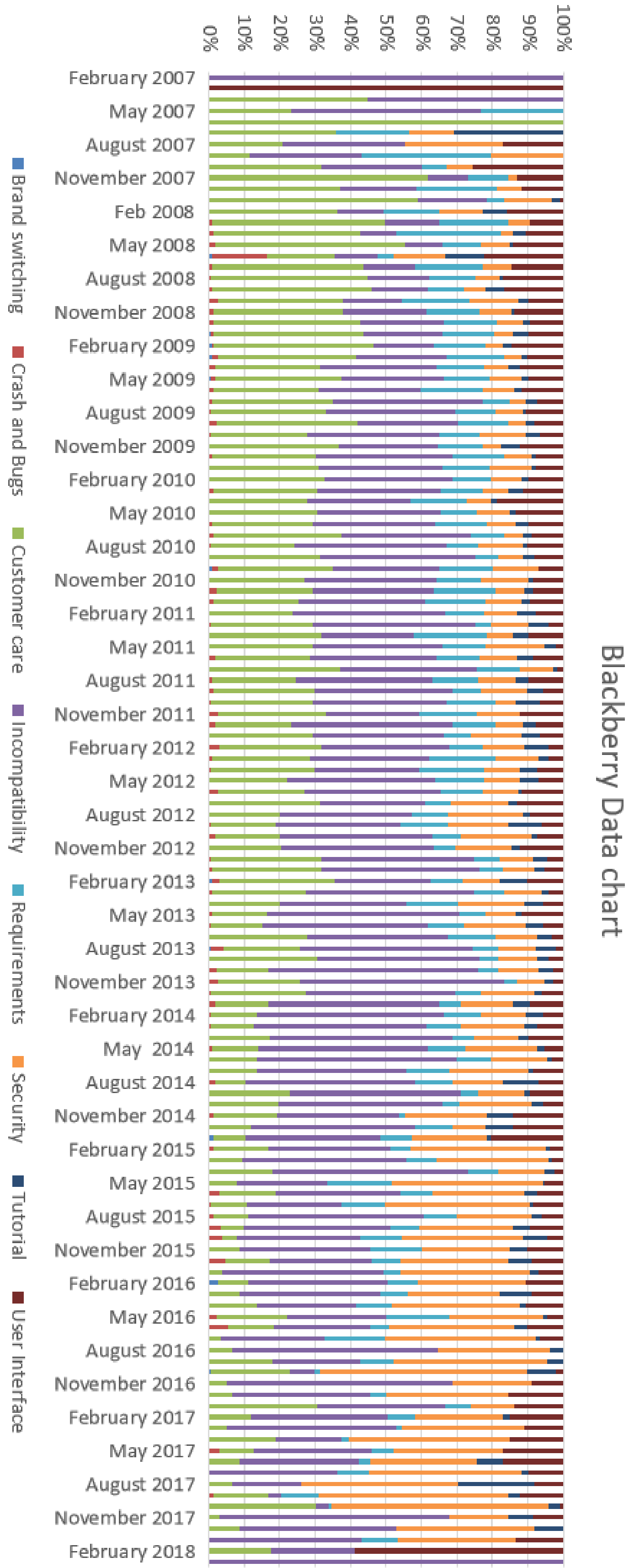


Figure A.3: Node's data chart of Blackberry between 2007 and early 2018

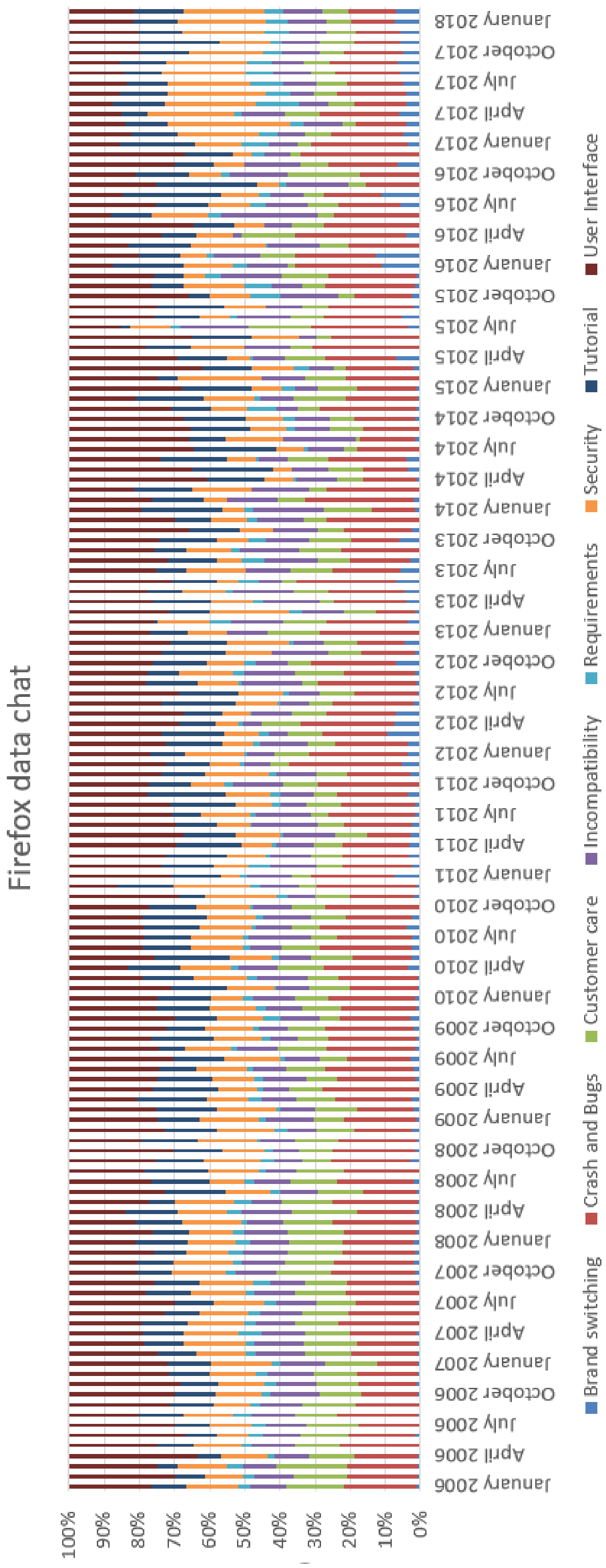


Figure A.4: Node's data chart of Firefox between 2006 and early 2018

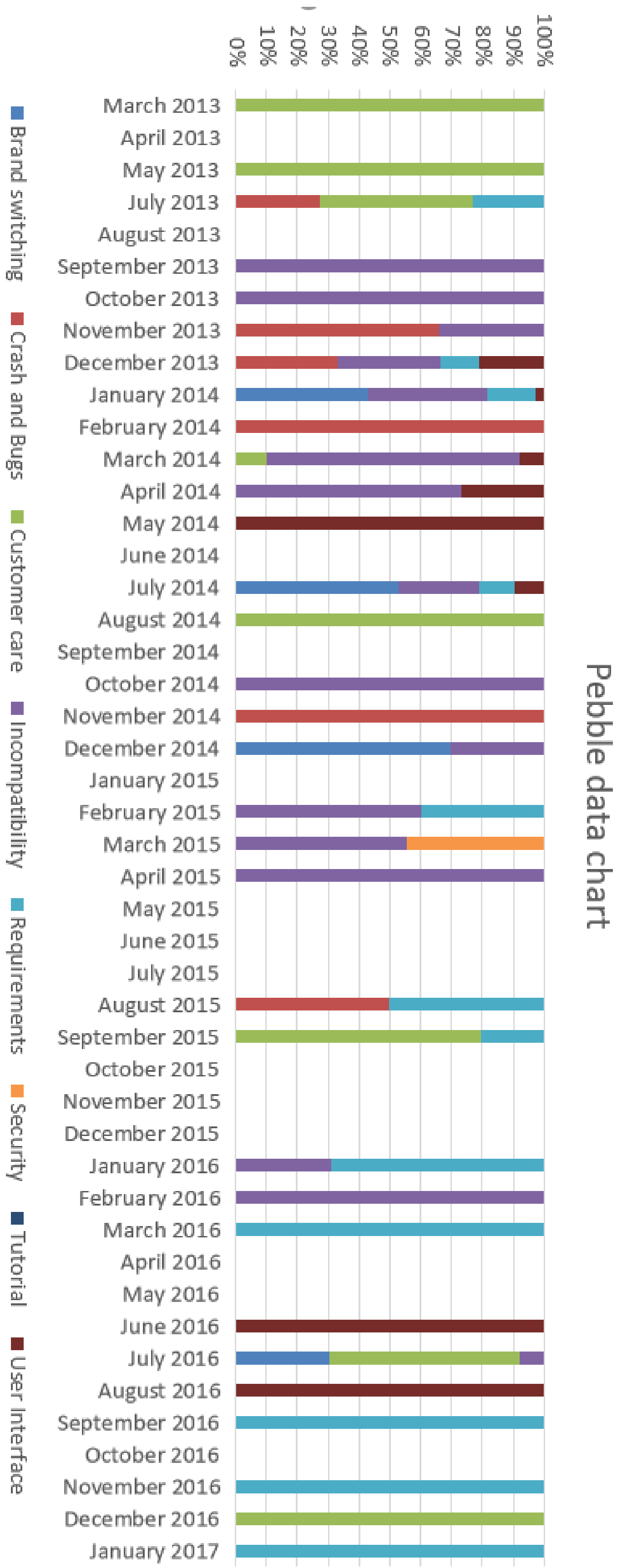


Figure A.5: Node's data chart of Pebble between 2013 and early 2017

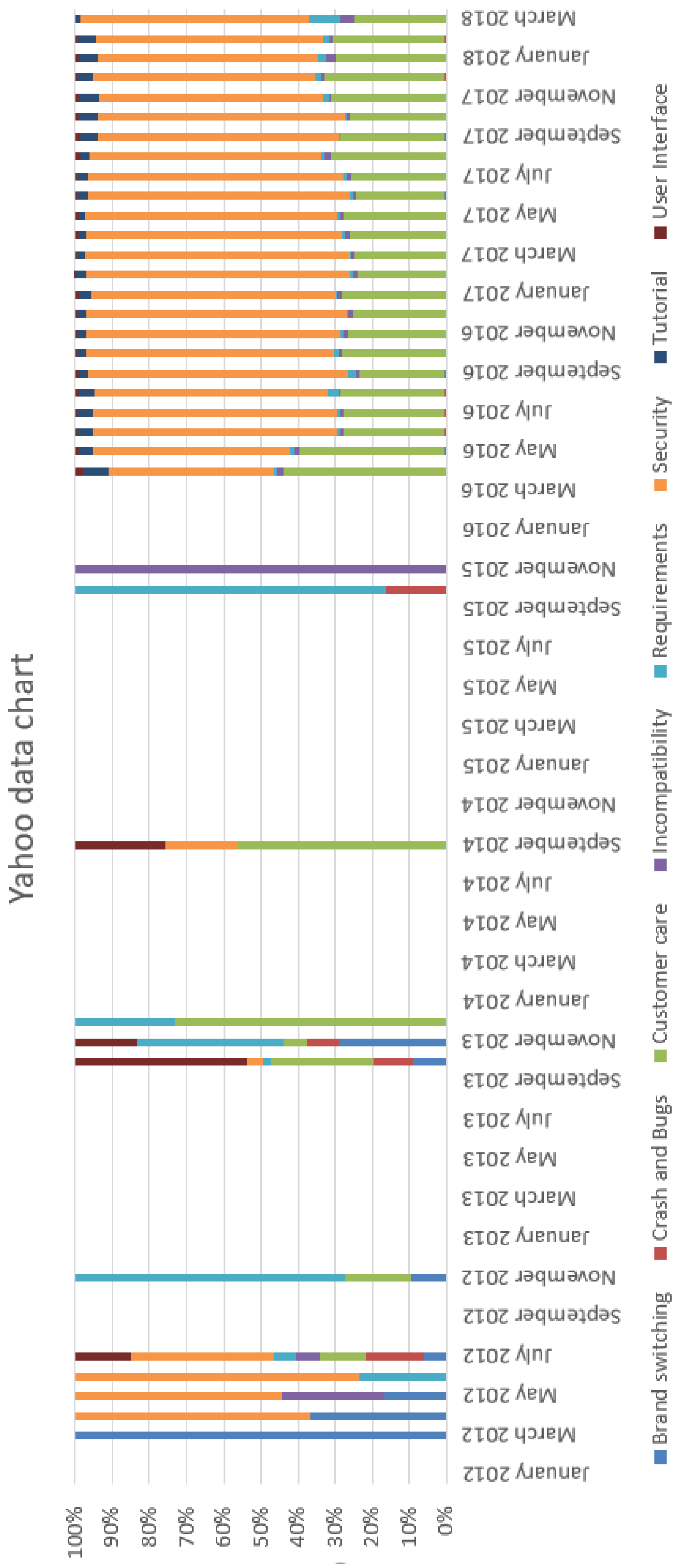


Figure A.6: Node's data chart of Yahoo between 2012 and early 2018

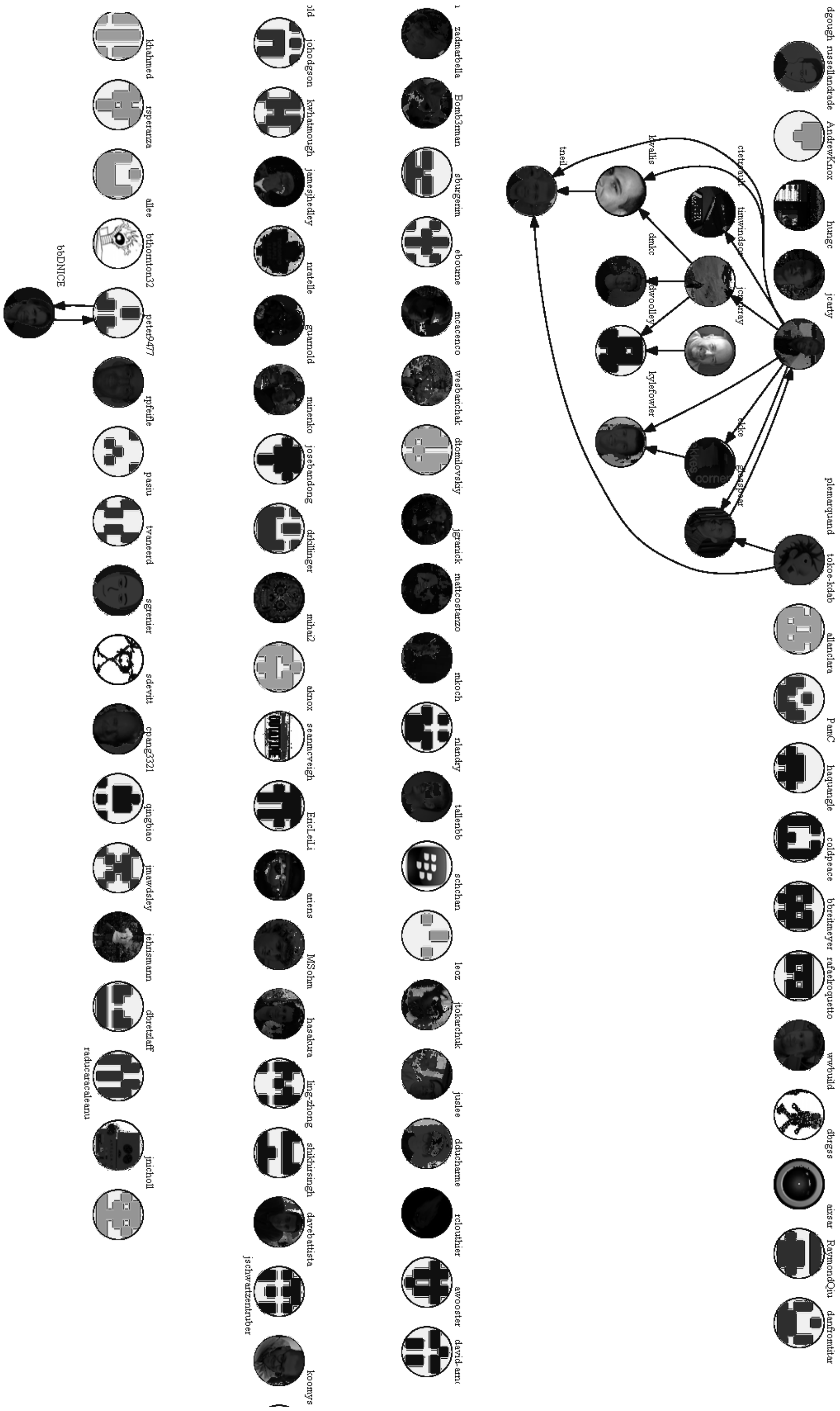


Figure A.7: Developer Network for Github users collaborating on the Blackberry Github profile and their connections to each other

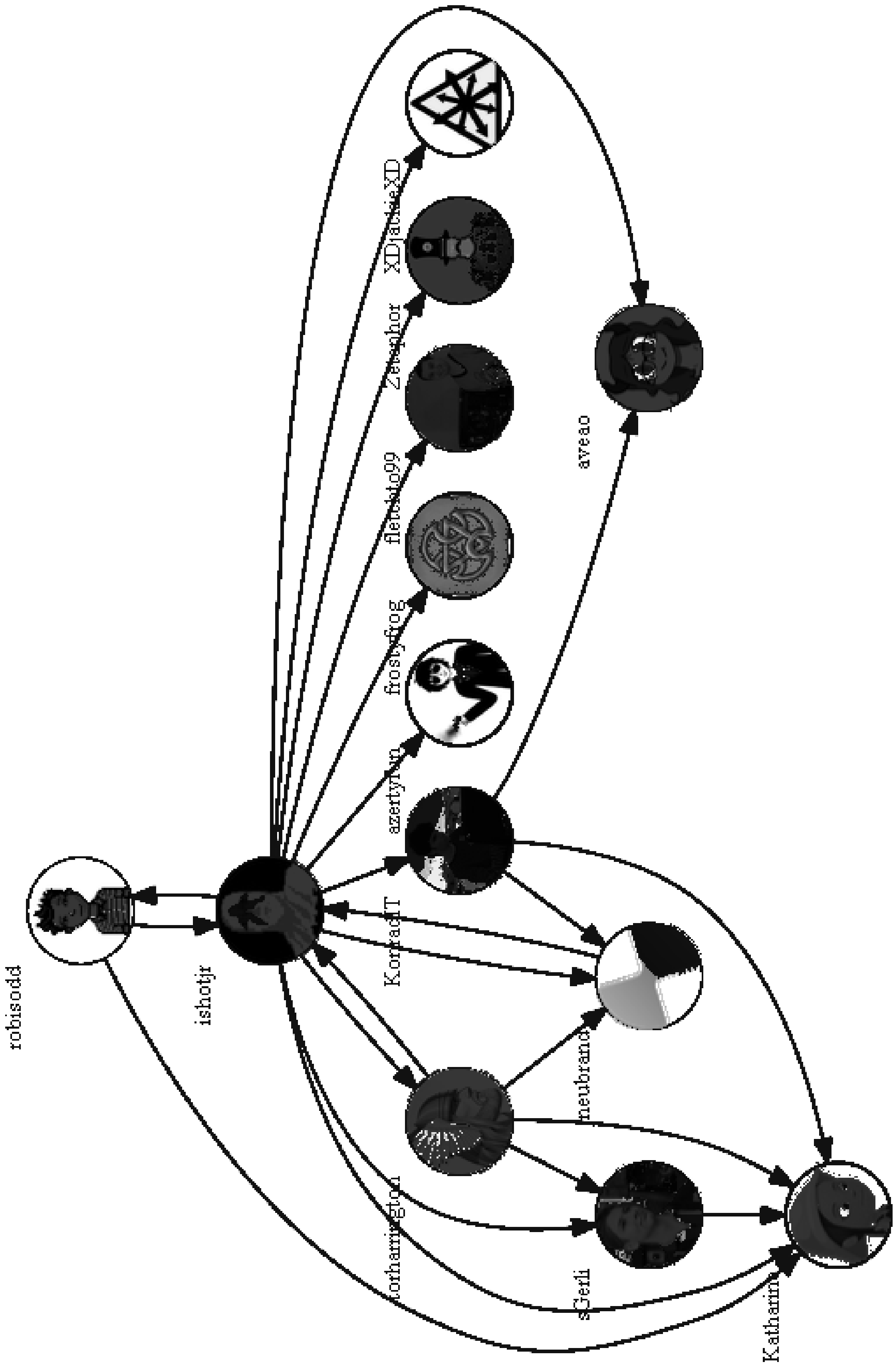


Figure A.8: Developer Network for Github users collaborating on the non official Pebble Github profile and their connections to each other

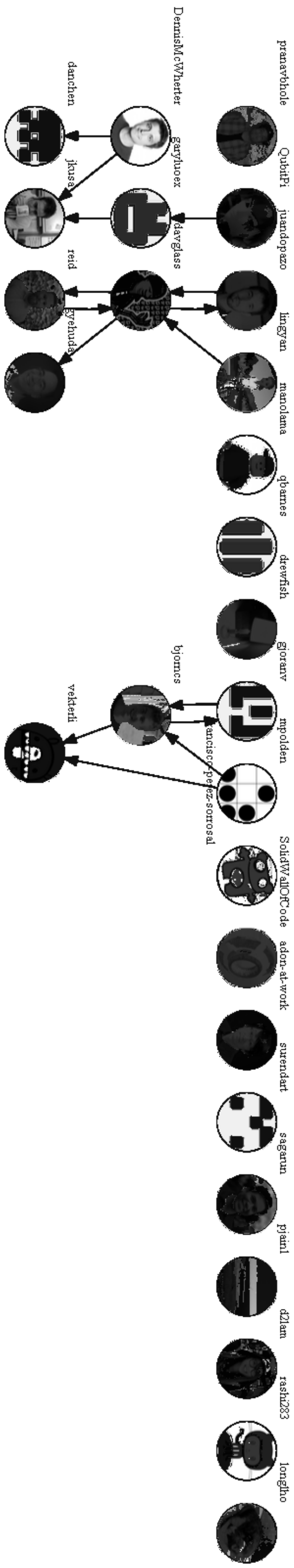


Figure A.9: Developer Network for Github users collaborating on the non Yahoo profile and their connections to each other

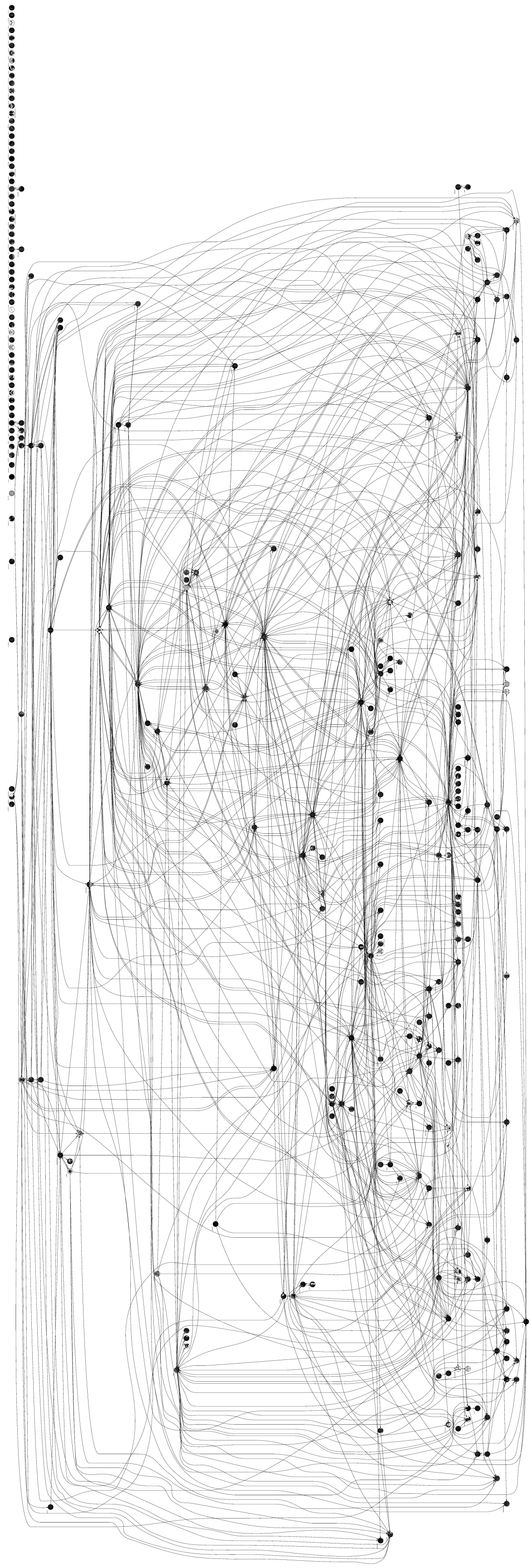


Figure A.10: Developer Network for Github users collaborating on the Mozilla Github profile and their connections to each other

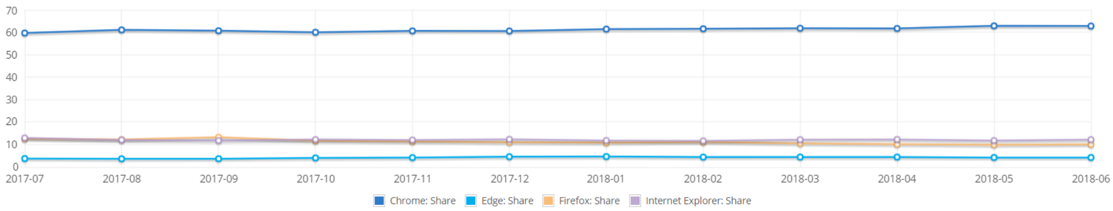


Figure A.11: Market share percentage of browsers between July 2017 and June 2018, retrieved from (NetMarketShare, 2018)

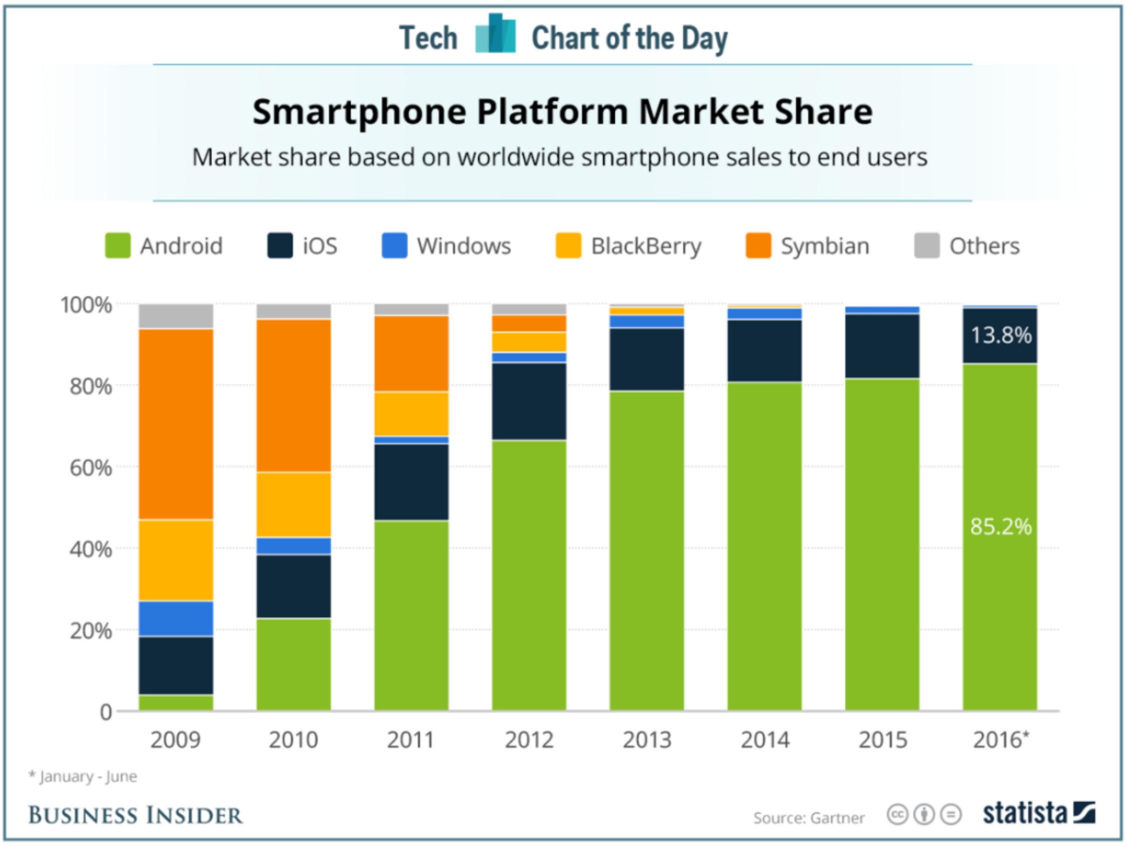


Figure A.12: Smartphone platform market share, retrieved from (Dunn, 2016)