Python: Characteristics Identification of a Free Open Source Software Ecosystem

Rick Hoving, Gabriel Slot, and Slinger Jansen
Utrecht University
Department of Information and Computing Sciences
Princetonplein 5, 3584CC, Utrecht, Netherlands
{r.hoving, g.c.a.slot}@students.uu.nl, s.jansen@cs.uu.nl

Abstract—Analysing a free open source software ecosystem can be beneficial and can help stakeholders in numerous ways. The analysis can help developers, investors, and contributors, to decide which software ecosystem to invest in and where to invest. Another reason for making an analysis is to assist ecosystem coordinators in governing their ecosystem. The paper provides an insight on the free open source software ecosystem of Python. It presents an analysis of the software ecosystem itself and the different characteristics it has. Based upon the conducted analysis with the available dataset, the research concludes that the free open source software ecosystem of Python contains three ecosystem roles that define its ecosystem. Next to that, it has grown exponentially from 31 active developers in 2005 to 5,212 December 2012. These results can help set up a strategy for the future of the Python ecosystem. At this point in time, it is necessary to make arrangements for the ongoing growth of the Python ecosystem. Failing to do so can lead to a growing number of unusable features, and eventually advance to an unhealthy ecosystem.

Index Terms—Free open-source software, Software ecosystem, Python, Exploratory case study, Roles, Characteristics

I. INTRODUCTION

Free open source software ecosystems (FOSSE) will insure a greater social welfare for its developers. To insure the social welfare, the FOSSE needs to be governed and improved[1]. In order to define a FOSSE, the paper builds upon the definition of a software ecosystem by Jansen, Finkelstein, and Brinkkemper [2]. This paper defines a FOSSE as:

“A free open source software ecosystem is a set of developers functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. The result of the interaction is freely available for everyone.”

According to Jansen et al. [2], it is important to know the dynamics of the ecosystem in order to govern and improve it. The profit of FOSSEs is not defined in terms of money, but in how well it operates and survives (i.e. how lively is the ecosystem). An analysis of a FOSSE can be beneficial and is necessary in order to generate the most profit. According to Kabbedijk and Jansen [3] and Jansen, Finkelstein, and Brinkkemper [4], specifying different features, like size, types of actors, roles, and connectedness of a FOSSE can help the developers, investors, and contributors understand the ecosystem. The insight will assist them in choosing the FOSSE to invest and where to invest. It can also assist the ecosystem’s coordinators in governing the FOSSE.

The different features of a FOSSE can be identified and has been done before by for example Kabbedijk and Jansen [3] and Jansen et al. [4]. Kabbedijk and Jansen [3] call the features of a FOSSE descriptives, while Jansen et al. [4] call them characteristics. Both the descriptives and the characteristics are used to indicate features of a FOSSE. Therefore both descriptives and characteristics are considered to be the same and will be called characteristics throughout the paper.

The goal of the paper is to see which characteristics can be identified within a FOSSE. In order to define its characteristics, the paper describes the data mining process and data analysis of the Python FOSSE. By defining the characteristics of Python, the paper helps further define the total set of characteristics that describe a FOSSE.

In the second section, a literature research is presented to see what characteristics have been defined for other FOSSEs. The section following up the literature research, section three, uses the information to state a number of sub-questions, which together will answer the main research question. Section four gives a description of the Python FOSSE together with the data mining process. Section five gives an analysis on the data obtained from the data mining process. After the analysis, the results section uses the analysed data to present the findings of the analysis. The last sections presents the discussion, conclusions and points for future research.

II. LITERATURE RESEARCH

Ecosystems have been studied within various researches. Iansiti and Levien [5] state that an ecosystem revolves around a platform. In a software ecosystem (SECO), the platform is usually a commonly used software product. However, according to Jansen et al. [4], a SECO is more than just the sum of its parts. It consists out of actors, such as independent software vendors, customers, resellers, and/or outsources. The dynamics and identity of a SECO can be defined by certain characteristics. Examples of characteristics are size, types of actors, roles and connectedness. As depicted in Figure 1, SECO models have three scope levels. Each scope level has its own study of subjects, and therefore its own characteristics.

The analysis of the FOSSE in this research will be carried out on the second level: the SECO level. Jansen et al. [4] state
that the SECO on this level can be defined by the internal characteristics. The internal characteristics show the general information of the SECO, amongst which are the size, in terms of numbers of the different types of elements within the SECO, the most important elements, and the connectivity between the different elements. Next to the internal characteristics, a SECO on the SECO level can be defined by the different ecosystem roles of the actors within the SECO, and lastly by its growth or evolution over time.

A more complete view of a SECO with its components is given in Figure 2. The figure is based on the domain model by Berk, Jansen, and Lützen [6] and extended using the findings of other studies. These other studies are further elaborated in the following paragraphs.

According to Iyer, Lee, and Venkatraman [7], there are three parts that are important in most networks: the Hub, the Broker, and the Bridge. Iyer et al. describe hubs as being firms with a disproportionately high number of links. Brokers are firms that creates a connection between two other firms. Brokers can be further categorized by a number of roles. Hanneman and Riddle [8] distinguish five different roles: Consultant, Coordinator, Gatekeeper, Liaison and Representative. Lastly, bridges are links critical to the overall connectedness within the network. According to Jansen et al. [4], bridges are mere properties of roles than actual parts of an ecosystem. In order to use the definition of Iyer et al., it is necessary to translate them to a FOSSE definition. Hubs can be translated to developers with a high number of links to other developers. Brokers and bridges also need to be translated in the prior stated manor.

The players within an ecosystem can be described as being a Keystone player, a Niche player, or a Dominator. Like the hubs, the definitions of these players are applicable for firms within a market. The translation of the players has already been done by Jansen et al. [4]. As stated by Iansiti and Levien [13], niche players add parts to the keystone platform, which is created and maintained by the keystone players. The conducted research is done based on so-called Python eggs. The focus of the paper is therefore on the niche players, who develop the Python eggs as add-ons to the actual Python language as being the keystone platform.

According to Hagel, Brown, and Davison [9], the niche players can be one of the following: an Influencer, a Hedger, or a Disciple. Although these roles are relevant for the completeness of the model, these roles will not be further studied. The roles are applicable for the first SECO scope level, which is beyond the scope of the research.

Kabbedijk and Jansen [3] studied the Ruby FOSSE and specified three roles: the Networker, the One Day Fly, and the Lone Wolf. The networkers are developers in the ecosystem that cooperates with other developers and play a large role in the SECO in terms of downloads. The one day flies are developers who only made a single contribution to the ecosystem. The lone wolf is a developer that has produced important parts for the ecosystem, but has no connections with other developers. As stated in the introduction, the goal of the paper is to see which characteristics can be found within a FOSSE. On the second SECO level, the research will search for the three specified roles: the networkers, the one day flies, and the lone wolves.

Next to the different elements that take part within a SECO, the ecosystem can be defined by health characteristics. The health characteristic is an important indicator of an ecosystem and has been studied in multiple researches, including by Den Hartigh, Vissher, Tol, and Salas [10], Wynn [11], and Iansiti and Richards [12]. A complete set of health characteristics has yet to be created or translated for the purpose of measuring the health of a FOSSE. Therefore, the research for the health characteristic will only focus on the growth and evolution over time.

### III. Research Questions

The goal of the paper is to identify the elements, properties and characteristics of a FOSSE. The main research question the paper answers is described as:

**RQ** What are the defining characteristics of a large scale FOSSE?

The research question will be answered using a number of sub-questions. Knowledge of the elements within a FOSSE is needed in order to conduct an analysis of the ecosystem. Only then can the analysis help stakeholders and coordinators. This leads to the following sub-question:

**SQ1** What elements exists within a FOSSE?

As derived from the literature research, a FOSSE can be defined by a number of other characteristics: the internal characteristics, the different roles of the actors within the ecosystem, and lastly by its growth or evolution over time. This leads to the following three sub-questions:

**SQ2** What are the characteristics of a FOSSE?

**SQ3** Which roles are played by the elements within the ecosystem?

**SQ4** What characteristics does a FOSSE have when looking at its growth and evolution?
IV. CASE DESCRIPTION AND DATA GATHERING

Python is an interpreted, interactive, object-oriented, open-source, and high-level general purpose programming language. It provides high-level data structures, such as lists and associative arrays, dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc. [14].

Python appeared in 1991, designed by the Dutchman Guido van Rossum and developed by the Python Software Foundation. To extend the Python language, developers are able to create packages, which are called Python eggs [15]. No restrictions lay on the number of eggs that a developer can create. The eggs can be created by developers that work alone or in collaboration with other developers. The dataset showed that the Python eggs first appeared in 2005. A Python egg is described as a logical structure embodying the release of a specific version of a Python project, comprising its code, resources, and meta data.

Because of the flexibility of the language, the Python eggs do not have to be written in the Python language. The eggs can also be written in the programming languages C, C++, Java using Jython, and .NET using IronPython1. After creating an egg, it can be uploaded onto the official Python webpage2, where it is stored into a database. In order to utilize an egg in a specific project, the egg needs to be downloaded from the prior stated website and imported into a specified project.

The Python eggs are mined from the Python website3. The dataset of Python was mined on the 3rd of December, 2012, using a custom made .Net script. The script is divided in two phases, the mining of the Python eggs and the mining of the developers of the Python eggs. Both phases populate a database with the gathered data.

The first phase uses HTML scraping in order to get the eggs from the list of Python eggs from their website3. A single egg contains a name, a description, and a link to the detail page of the egg.

The second phase does the collecting for the links that are gathered in phase one and uses HTML scraping to collect the developers that created an egg, together with the number of downloads of that particular egg. The user-names stated as the “Package Index Owner” are defined as the owner of a Python egg. Because these user-names are not the real names of the developers, their identity cannot be identified. Therefore it is impossible to identify persons like for instance Guido van Rossum.

Another activity within the second phase is splitting the data into separate developers. This is done using an SQL script. As an example: an egg specifies "kös, ejucovy" as being the developers of the egg, "kös, ejucovy" is split into "kös" and "ejucovy" and puts the split data into a database.

An overview of the results of the data mining is shown in Table I. Figure 3 represents the visualization of the mined Python network.

1http://www.python.org/about/
2http://pypi.python.org/pypi/
3http://pypi.python.org/pypi/?%3Aaction=index

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique developers</td>
<td>9,189</td>
</tr>
<tr>
<td>Unique eggs</td>
<td>27,624</td>
</tr>
<tr>
<td>Unique links between developers</td>
<td>8,893</td>
</tr>
<tr>
<td>Total links between developers</td>
<td>93,770</td>
</tr>
<tr>
<td>Developers without links</td>
<td>7,157</td>
</tr>
<tr>
<td>100% of all the links</td>
<td>Between 20.11% of the developers</td>
</tr>
<tr>
<td>Number of developers that created one egg</td>
<td>4,992</td>
</tr>
</tbody>
</table>

TABLE I
CHARACTERISTICS OF THE PYTHON FOSSE.

Fig. 3. The visualization of the Python ecosystem. The different developers are represented by the nodes. The size of the nodes indicates the number of links the developer has. The bigger the node, the more links the developer has. The figure is formatted in a way that all the groups of developers are packed together. The grouping is done using the Markov Cluster Algorithm[16]. The groups are colour coded, i.e. every group has its own colour. The developers are linked together when they worked together on the same eggs.

V. ANALYSIS

Two distinctive elements can be found within the FOSSE of Python. Namely, the Developer and the Egg. The Developer element represents the different developers that have an account on the Python website and have created at least one Python egg. Developing, updating or fixing bugs for the existing Python eggs are the actions that a typical developer would do, whilst participating in the Python FOSSE. The Egg element represents the Python egg. As stated in the case description, the Python egg is an extension on the Python language and can be freely downloaded and used together with other eggs.

As can be seen in Figure 4, the elements have a number of distinct relations between them. The properties marked with an asterisk (*) are optional properties.

Within the Python FOSSE, the following roles have been identified: the one day fly, networker, and the lone wolf.

One day flies: The one day fly is a single developer who has made one egg by itself and the number of downloads of the egg is in the top 5% of most downloaded eggs. 5% is chosen to be able to compare the results of the paper to the results from Kabbedijk et al. [3] whom specified the same 5%. The information could not be mined from all the eggs, because

Image 330x401 to 546x617
the number of downloads is an optional property. From the total of 27,624 eggs, only 23,493 eggs hold data on the number of downloads. Therefore, the following information is based on the subset of 23,493 eggs. The list of the top 5% of most downloaded eggs contain 1,174 eggs, ranging from 3,975 downloads for the "Machine learning and interactive data mining toolbox" egg to 2,214,509 downloads for the "System for managing development buildouts" egg. The list of one day flies in the top 5% most downloaded eggs contain 60 developers. A subset of the list is presented in Table II. The one day flies in the list range from 4,048 downloads, for the "A python-implementation of an Erlang node" egg, to 447,100 downloads, for the "Amazon Web Services Library" egg.

Networkers: As stated before, the networker is defined as a developer who plays a big part in the total number of downloads of all the eggs it has written. If an egg is written together with other developers, the number of downloads counts for all the developers. A network of the top 40 developers is shown in Figure 5. Note that for the same reason, the subset of 23,493 eggs is used.

Lone wolves: The lone wolf is defined as a developer who, just like the networker, plays a big part in the total number of downloads of all the eggs it has written. The difference between a networker and a lone wolf is that the lone wolf has never cooperated with other developers. In the Python FOSSE, 5,716 lone wolves have been identified. Table III shows the top 5 of the identified lone wolves within the Python FOSSE. Note that for the same reason, the same subset has been taken as the two prior roles.

VI. RESULTS

One of the characteristics that defines a SECO is the growth or evolution over time. A number of results can be presented based upon the analysis of the dataset of the FOSSE of Python.

First of all, Figure 6 depicts the total number of eggs that have been added to the Python FOSSE each year. In 2005, 36 eggs were added to the FOSSE, in 2012, the number has increased to 12,063 eggs.

Figure 6 also depicts the active developers over time. As stated before, the dataset only holds data from 2005 till 2012. There were 31 developers who developed or helped develop a Python egg in that year. In 2012, the number of active developers has grown to 5,212 developers whom added a new egg to the FOSSE of Python.

---

**Table II**

<table>
<thead>
<tr>
<th>Developer</th>
<th>Egg Description</th>
<th># downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>garnaat ping</td>
<td>Amazon Web Services Library UUID object and generation functions (Python 2.3 or higher)</td>
<td>214,519</td>
</tr>
<tr>
<td>jgraham</td>
<td>HTML parser based on the WHAT-WG Web Applications 1.0 (&quot;HTML5&quot;) specification</td>
<td>168,099</td>
</tr>
<tr>
<td>gutworth</td>
<td>Python 2 and 3 compatibility utilities</td>
<td>149,957</td>
</tr>
<tr>
<td>noah</td>
<td>Pexpect is a pure Python Expect. It allows easy control of other applications.</td>
<td>91,252</td>
</tr>
<tr>
<td>jkbr</td>
<td>HTTPie - a CLI, cURL-like tool for humans.</td>
<td>4,255</td>
</tr>
<tr>
<td>nosexunit</td>
<td>XML Output plugin for Nose</td>
<td>4,203</td>
</tr>
<tr>
<td>Benjamin.Wilbur</td>
<td>Google analytics web property per site in django admin</td>
<td>4,168</td>
</tr>
<tr>
<td>jfennell</td>
<td>sqlite-backed dictionary</td>
<td>4,163</td>
</tr>
<tr>
<td>ketralnis</td>
<td>A python-implementation of an Erlang node</td>
<td>4,048</td>
</tr>
</tbody>
</table>

**Table III**

<table>
<thead>
<tr>
<th>Developer</th>
<th># eggs</th>
<th># downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>adustman</td>
<td>3</td>
<td>726,743</td>
</tr>
<tr>
<td>euske</td>
<td>3</td>
<td>595,292</td>
</tr>
<tr>
<td>garnaat</td>
<td>1</td>
<td>447,100</td>
</tr>
<tr>
<td>barryp</td>
<td>2</td>
<td>409,439</td>
</tr>
<tr>
<td>jamesbowman</td>
<td>4</td>
<td>283,933</td>
</tr>
</tbody>
</table>

**Fig. 4.** Elements within the Python FOSSE together with the relation between them. *not required property

**Fig. 5.** Top 40 developers in total number of downloads. The links represents the developers working together to create an egg. The more developers a developer has cooperated with, the larger the node.

**Fig. 6.** The number of eggs added to the Python FOSSE (blue line) and the number of developers who added an egg to the Python FOSSE (red line) per year between 2005 and 2012.
FOSSE.
Gems in the Ruby FOSSE and the Python Eggs in the Python FOSSE. Note that "add-ons" refers to the Ruby FOSSE by Kabbedijk and Jansen [3], and the data found in V gives a comparison between the data found within the Ruby FOSSE by Kabbedijk et al. [3]. Table language.

because all eggs are add-ons to the Python programming language. Figure 7 depicts a trend of the growth of the number of downloads. The number of downloads is the total number of downloads of every egg in the Python egg list and shows the new downloads of every year. The number of downloads for each year starts in 2005 at 105,784 and ends in 2012 with 22,461,378 downloads.

The frequency of used categories give a good insight on which type of eggs are mostly created. The top 5 of the most used categories are presented in Table IV. As can be seen in Table IV, most eggs are categorized in the category "Programming Language: :: Python". The category is logical, because all eggs are add-ons to the Python programming language.

The same three roles were found in the Python FOSSE as identified in the Ruby FOSSE by Kabbedijk et al. [3]. Table V gives a comparison between the data found within the Ruby FOSSE by Kabbedijk and Jansen [3], and the data found in the Python FOSSE. Note that "add-ons" refers to the Ruby Gems in the Ruby FOSSE and the Python Eggs in the Python FOSSE.

VII. DISCUSSION

The data set of Python was not mined without complications. For example, a number of properties of the eggs are not mandatory to fill in. This resulted in a data set where only 23,493 eggs of the total of 27,624 eggs hold data on the 'Number of Downloads' property. The dataset that was used for the analysis of for number of downloads was therefore slightly different from the dataset used in other cases. For instance, the comparison of 'Number of Downloads' and 'Uploaded-on' could skew a comparison. It is logical that some properties of the eggs are optional, like the 'Requires' and the 'Maintainer'. On the other hand, some optional properties did not make sense being optional, like the prior mentioned 'Number of Downloads' and 'Uploaded-on'.

As explained in the paper, the 'Requires' property presents the egg(s) needed, in order for the specified egg to be able to work. The property can be chosen with the creation of a new egg item in the Python egg list. However, the data mining results did not show these dependences. The analysis for the interdependencies of an egg is therefore left out.

Only a part of the characteristics of the FOSSE have been analysed. The brokerage roles (Consultant, Coordinator, Gatekeeper, Liaison and Representative) and health characteristics need additional research. The brokerage roles can be studied by means of a questionnaire in order to find the reason why developers have been working together and how they started the working relationship. As explained in the literature research, the health characteristics need to be translated in order to use them in the measurement of a FOSSE. Therefore, a complete image of the ecosystem of Python has yet to be given. However, the data presented in the paper does give a good overview of what is happening in the Python FOSSE.

VIII. CONCLUSION

By looking at the number of eggs created and number of downloads, developers within the Python FOSSE are able to see which developers have the most knowledge. Therefore, developers can see where to go to with their complications, which will results in a more social interaction between developers in the FOSSE of Python. The paper is comprehensive in giving economic advice to the stakeholders. For example, investors will have a better view of where to invest. Other than economic advice, the insight can help the FOSSE grow to a more profitable ecosystem. As an ecological point of view, the paper presents a better understanding about the Python FOSSE. Thus, making it more clear for the FOSSE’s coordinators on where to improve the ecosystem.

What elements exists within a FOSSE?: Within the FOSSE of Python, two elements have been distinguished. The found elements are the Developer and the Egg element. Each element having their own properties.

What are the characteristics of a FOSSE?: Within the FOSSE of Python, 9,189 developers were identified with the mined data set. These developers have created a total of 27,624 eggs. The total number of eggs show a total of at least 65,324,834 downloads. Over 450 categories can be associated with the egg in order to classify the eggs. With the most used category being 'Programming Language: :: Python', which has been used 15,287 times. In comparison with the data of the Ruby FOSSE, it can be concluded that the Python FOSSE has more than double the development Ruby has with almost the same amount of relationships between the developer. Therefore, the developers of Ruby are better intertwined with each other, in comparison with the developers of Python.
Which roles are played by the elements within the ecosystem?: Looking back at Figure 2, the paper set out to find three niche player roles within the Python FOSSE: the one day fly, networker, and lone wolf. All three of the roles are found in the FOSSE of Python as presented in the paper.

What characteristics does a FOSSE have when looking at its growth and evolution?: A number of characteristics can be identified when looking at the growth of the Python FOSSE. These characteristics are the ‘Number of Active Developers’, the ‘Number of Downloads’, and the ‘Number of New Eggs’. As shown by the presented data, the FOSSE of Python is growing rapidly, each year being better than the last.

Improvements of the Python FOSSE: A number of key aspects can be identified according to the definition of a FOSSE. Namely, the developers working together as a group, the shared market for software and services, and the relationships between them. For the Python FOSSE, the shared market for software and services and the relation between them is the same for every developer. Therefore, no improvements can be made in this area. However, the number of relationships between developers that have been working together on eggs can be augmented. The results indicate that within the FOSSE of Python roughly 22% (2,032 out of 9,189) of developers have cooperated at least once with another developer. For the developers to act as a more integrated whole, the 22% must be increased.

As stated, the Python FOSSE is experiencing a rapid growth in numbers of developers and eggs. This growth rate is a good indication of its health. However, it should be kept in mind that the current facilities will have to handle a much higher load when this trend continues. Python’s strategy might need to be altered in order to handle the increasing load.

According to Corallo [17], diversity is a key enabler of the ecosystem. One may argue the similarity to natural selection. When an environmental shock occurs in the Python FOSSE, for instance with a breaking update, all the developers will have to update their eggs. Such an environmental shock results in a lot of eggs being not relevant to the FOSSE any more. In order to properly handle the shock, Python should be prepared for it and handle accordingly by removing the irrelevant Developer and Egg elements in the FOSSE.

IX. Future Research

There are some parts in the area of characteristics of the Python FOSSE that have not been covered in the paper, but are worthy of being researched in the future.

The research of the paper focuses on the Python FOSSE. Looking at the roles played in the Ruby FOSSE (one day flies, networkers, and lone wolves)[3], it is clear that the same roles can be identified in the Python FOSSE. However, these are not the only two FOSSEs and it is therefore not clear whether the characteristics in other FOSSEs are the same as well. A research focusing in this area could search for similarities and/or differences in characteristics between the Python FOSSE and other FOSSEs.

Another area is to search for differences between different kinds of SECOs. The paper focuses on the FOSSE, but it is not clear whether the FOSSE characteristics differ from other kinds of SECOs.

The study of the Python FOSSE focuses on the SECO scope level (2), which studies the software supply networks and their different relationships. This is however, just one of the three different scope levels. Future research can look into the other scope levels and search for characteristics of a SECO there.

Researching the brokers and bridges of an ecosystem would show the reason why firms, or developers, have a connection with each other in the second scope level. As explained in the discussion section, this can be done by conducting a questionnaire.

Like the above characteristics, a FOSSE’s growth or evolution over time is a characteristic that defines a SECO. The period for the growth in the paper was from 2005 until the 3rd of December, 2012, in which Python was doing better each year. Future studies can look at a later period and see whether the results presented in the paper are ongoing or that the trend will change over time.

References