AN ABSTRACT OF THE THESIS OF

Rithika Kiran Naik for the degree of Master of Science in Computer Science presented on September 17, 2015.

Title: Visualizing Contribution Patterns in Open Source

Abstract approved: _____

Dr Carlos Jensen

Open Source software gives users the freedom to copy, modify and redistribute source code without legal entanglements. The evolution of these software communities usually depend a lot on how the participating developers and users interact and co-operate with each other. Over the past few years, open source software have become widely accepted and used, hence making the study of their organization and sustainability a very hot topic. To understand the working philosophy, health, and evolution of any open source project, one needs to look at a variety of factors like community interaction, strength of connections, recruiting and mortality, support and contribution patterns. Getting, sharing and analyzing Software Development metrics and information related to open source projects has thus become important, with many projects and companies looking to do so. The main challenge however is distilling large amounts of data into actionable intelligence. The main idea behind this work is to help communities and others understand con-

tribution patterns in open source projects through the use of various visualization techniques.

Visualizing Contribution Patterns in Open Source

by

Rithika Kiran Naik

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented September 17, 2015
Commencement June 2015

Master of Science thesis of Rithika Kiran Naik presented on September 17, 2015.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Rithika Kiran Naik, Author

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my advisor, Dr Carlos Jensen, for his patience and constant support throughout the phase of this research. All the brainstorming sessions to discuss every minute detail in progress were extremely fruitful. My sincere gratitude to Dr Anita Sarma, Dr Cindy Grimm, Dr Cathy Mullet for accepting my invitation and taking time to be a part of my Graduate committee.

My gratitude to team Bitergia as this work would not have been possible if not for their quest to learn more. So thank you Jesus M Gonzalez Barahona and Gregorio Robles for providing me with the necessary datasets and valuable suggestions.

Thank you Shankar Jothi and Subarna Kar for all the help during the course of this work and all my friends in Corvallis for the wonderful 2 years away from home. Special thanks to the entire Human Computer Interaction group for looking through my work and providing me with timely inputs.

My dream of pursuing a Masters degree would not have been fulfilled if not for my parents constant encouragement and faith in me. Thank you Mom and Dad for everything !

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

The software industry has undergone a lot of changes over the years and one such important change was the advent of Open source software. Open source software is a type of software development which is volunteer driven [14] and the code base is usually made available for any person to modify, edit and redistribute [2]. Open source software is being heavily relied on and used today because of, one its quality.Would you prefer a software written by a bunch of developers or software written and improvised by thousands of developers? It follows the principle of "Given enough eyeballs, all bugs are shallow" [8] where it is found that in open source software as a large base of developers contribute to the code base, this would only lead to a large variety of features, huge support groups and enhanced products[3]. Second would be its flexibility where in, the users can tweak any part of the functionality to match their needs and this is because the code is open and available. Third would be its auditability where one can believe the claims made by the authors about issues like security, freedom, flexibility etc which is highly lacking in closed source software. Next would be the large support base which is available in the form of documentation, forums, mailing lists, forges, wiki, live chats and so on and with so many options available a user can never be holding onto to a problem ling enough. With a plethora of such important features, the need to use open source software has hence being going uphill.

Managing any software comes with its own set of challenges and open source software is no different.Having mentioned its advantages, handling them and keeping to the standards is a big task. With the code base made available to all, it brings out issues like maintaining coding standards, peer review processes, skillsets, documentation etc [4]. The fact that its volunteer based brings out issues like community management, fairness and equity among the developers. Open source software relies heavily on its community for its growth and development [7] making co-ordination among them a foremost point of importance. The OSS community usually has a lot of developers collaborate from all over the world who work voluntarily to produce code and support the software in general. Their motivations and ethics for this have always been open for studies [9]. There are also other reasons that define if a developer would want to contribute to the project in the first place. Some such reasons would include their education levels, facilities available, time commitment, the need to use the project, code interaction, community support, management and governance to name a few [10]. Unlike traditional softwares, as open source software has a wide spread developer/volunteer group, the success or failure hence depends strongly on its community as they can make or break the project. Their motivations and interests are often the reason sometimes for the birth of a project. A very motivational discovery in this area was the onion model which clearly divided the developer lifecycle into 7 layers, with each layer moving inwards, showing the level of involvement of the developers. The outer most layer is considered as the starting point for a new developer after which he slowly transitions to higher points of responsibilities if he continues to stay with

the organization, with the final stage being becoming a part of the core community and a regular contributor[6].

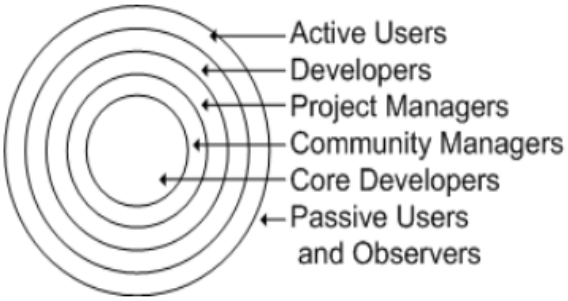

Figure 1.1: Onion model by Ye and Kishida

This being the basis to understand a developers growth, how can one know the spread of the developers contributing? Current approaches rely on very biased or incomplete knowledge bases like for instance, finding a large, new developer base and marking that to be as a growing project [16]. As the community is formed by developers from all across the world, it becomes important to know which regions contribute more to the open source world and which regions need help. This can help researchers to get a better feel of why developers in a certain region contribute more or less than others. This was our first motivation for research. Also a projects peak phase of success can be noted by looking at the community size and contribution size at any given time. This would give a clear picture about a project's good phases and bad phases. Such studies often help as we can help find a future course of action for projects dwindling from their paths so they can retain their community and don't just vanish from the face of the open source world. This became our second motivation.

1.1 Why Data Visualizations

In 1973, the Anscombe's Quartet showed how four sets with nearly identical statistical properties but appeared very different when graphed [19]. This led to a serious thought that understanding data by just looking at it would not be useful or correct. Hence visualizations were given a lot of importance. As the proverb says,"A picture is worth a 1000 words", visualizations made headlines for having thrown light on datasets in a completely new way. As the amount of data will only increase in the near future with each of it having a different meaning, a common way of unifying the data with the its underlying meaning being intact was much needed. This is how data visualizations became popular. This is the very reason why we decided to work with them as well. Comparing, contrasting, reviewing data becomes easier when one views various visualizations. But creating visualizations is not enoguh, one needs to understand the requirement in hand, perform data manipulation accordingly and then create a suitable visual diagram. Because if we cannot differentiate between excess data and useful data, the whole point of using visualizations for analysis is lost.

The main goal here was hence to build a neat, easily accessible, intuitive and effective set of visual interfaces which can give an insight on the trends in an open source project with regard to its developers contributions and experience.

Chapter 2: Related Work

Researchers all over the world have performed different types of studies, surveys, observations etc on the trends in open source projects. They are continuosly looking at the open source communities and contributor motivations to get a grasp on how the open source communities function to be sustainable. For this very reason Linux Foundation releases their survey study to understand how they performed each year and looks at various entities ranging from code quality to developer base to support issues and their preferred way of showing these varied results are through visualizations [20]. This has become like a guideline for most surveys today as it has a lot of information that would help understand the growth of an open source community. Survey's have been conducted with around 1600+ developers all over the world to look at areas like developer role in OS projcts, extent and intensity of OS contributions, support coming in by core team. The survey also showed some demographic data like male to female ratio, employment status, living area, formal education level, reasons to contribute to an open source project etc [17]. But the survey just showed a general overview of developers contributing and made no emphasis beyond that. Another interesting area looked at to understand a developers motivation was to understand how their roles affected the evolution of the community. Ye and Kishida looked into these motivation fcators and stated that learning plays a big part in getting developers to contribute [6].

This was seconded by the study to understand the development in Apache and Mozilla, where they stated that learning and capability to solve a problem are major motivations for developers at any level in an open source project [18]. A lot of research has been done to understand developer motivations in open source and sustainability factors like number of developers in a project [17, 26], size and quality of the code base [28], number of users for any given project [26], functioning principle of projects [27], etc.

In 2014, research was done to gauge the health of an open source community. This work revolved around the fact that health assessment was done by tools prepared to gauge certain communities or platforms but were not universal in nature for example, specific solutions existed for communities like GNOME [30], Debian [29] etc by accessing the data repositories but the same tools could not be used for other communities [30]. There are frameworks that monitor and analyse repositories and is more generic but still has its own limitations with technologies [28]. This research made use of service oriented computing monitoring techniques to gauge the health of an open source community by being as generic as possible and looked into the code base and activity revolving around it [28]. The evolution of open source projects was keenly looked upon by looking at methodology, code sets, tool sets, performance evaluation, maintainability, social network growth of developers and many such attributes that have an impact on a community [27].

Most researchers found one thing common when it came to gauging health and evolution of open source software and that was, the impact that communities had on the projects. It became very clear that a sustainable community can be

attained only when a constant inflow of developers with varied skills, ages and from a geographically wide area are present. This would been a wide spread knowledge about the project and a good mix of ideas and interests. We hence decided to narrow down our focus to understanding factors like age/experience of developers and their locations so we can bring out a better idea about their workings and the effect of these in the community on the whole.

2.1 Experience of Developers

In 2002, a study was conducted on two open source projects,Apache and Mozilla,to understand open source development. One of the parameters that was looked at was developer participation where they looked at the frequency of contributions over a period of time. They concluded that very few developers who are not a part of the core group contribute regularly in Apache and Mozilla showed that the core group was very active as well[18]. The professors of Notre Dame University studied the parameter on a larger group of varied open source projects for better understanding of open source project development. They noted that developers in open source projects are not as well connected as developers in other collaborative networks. Also developers were more in number in general in larger more successful projects than in smaller fairly unknown projects [9]. Studies have been conducted to know the age range of developers, motivations of developers, but the focus on understanding the time aspect of how long developers stay in an organization and finding a pattern of action for the same is our area of interest as this will give

us an idea about the growth or downfall of an open source project. In an article in 2014, researcher Jesus states that in order to gauge a community's health it is important to keep track of the developer ages. He goes on to state that attracting new developers is important but retainig the older ones also is a high priority task. The ratio of experienced, long term members to recent ones gives us details about the quality of code and the need to support members [13].

Turnover: Shows how people are entering and leaving the community. An indication of the community's attractiveness.

Age: Measures how long ago each current member joined a project. Gives out how many people are available at different stages of experience, from old-timers to newbies.

Together these metrics can be used to estimate engagement, to predict the future structure and size of the community, and to detect early potential problems that could prevent the healthy growth of a project [13].

2.2 Geography of Developers

Its intriguing to note that though studies have been conducted to understand the motivations of the developers, understanding the range of developers contributing, be it geographically or age wise still has a lot of scope. Greogorio Robles and Gonzalez-Barahona were among the first ones to look into the geographic factors that influenced developer contributions. They collected data of all the registered users from sourceforge as it was the largest libre software web based platform which

had data about geographic locations. After performing several techniques on this data they came up with a distribution of developers in different regions of the world[1]. The problem here was that the information about location or geography was not stored clearly in sourceforge. After this came the study by Takhteyev Y, Hilts A and they studied open source data made availble by Github, which was becoming a pioneering source management software. But there were problems here as well as the dataset was either incomplete or was not categorizable by TLDs [5]. Then came the idea of looking at timezones and IP's of the systems of the users who contribute, both collected through sourceforge. But the issue with looking at IP addresses was that the dataset is not available to all and agreements needed to be signed for the same[12]. Studies involving looking at mailing lists data to understand developer geography has not quite seen the light of day yet. If the data can be accessed then the same studies could show effective results as well. Results from the above mentioned studies all show that North American and European countries particularly USA are the major areas where developers are mostly located and work from. A survey by Ghosh et al[14] showed the influence of French, American and German developers but the survey that was cnducted by Tuomi et al[15] showed varied results leaning a lot towards US, UK, Canada and Germany. If the area is further widened and looked up on continent wise then North America takes the lead followed by Europe and clusters of Asia, Australia, South america and Africa and this was from the study by Greogorio Robles and Gonzalez-Barahona on sourceforge data[1].

Chapter 3: Data Gathering and Processing

Bitergia was founded by four professors from the Universidad Rey Juan Carlos, Spain in July 2012 [35]. They were researchers who were involved in understanding the open source community and its growth for long. This was the main idea behind starting Bitergia, the software development analytics company. The focus here was to have data analytics techniques to mine information about project performance, track developer actions, find areas of improvement and identify risks involved in further development.

3.1 Services provided

As providing software metrics was the goal here, they started off by building tools that would help them achieve the same by extracting data from multiple forums. They built tools to extract data from repositories like svn and git, tools to for mailing list data, for bug trackers like Jira, Bugzilla, Allura etc. These bunch of tools together are known as the *MetricsGrimoire Library* which is accesible for all in Github.

Now that they had the tools ready to extract the data, the next step was to build a framework which would help them to analyze and then visualize the data which they would obtain from the MetricsGrimoire tools. This is known as the

Figure 3.1: Bitergia data forums

VizGrimoire Library. With such a vast set of tools available our first instinct was to use them to extract data. But upon talking to the researchers they gave us access to their global database of data for the organizations they had extracted data for. We went ahead with this idea as, one it gave us time to focus on the visualizations than spend time on extracting the data and second, the dashboards made by Bitergia have been highly acclaimed and used by organizations like Apache, Eclipse, Puppet Labs etc.

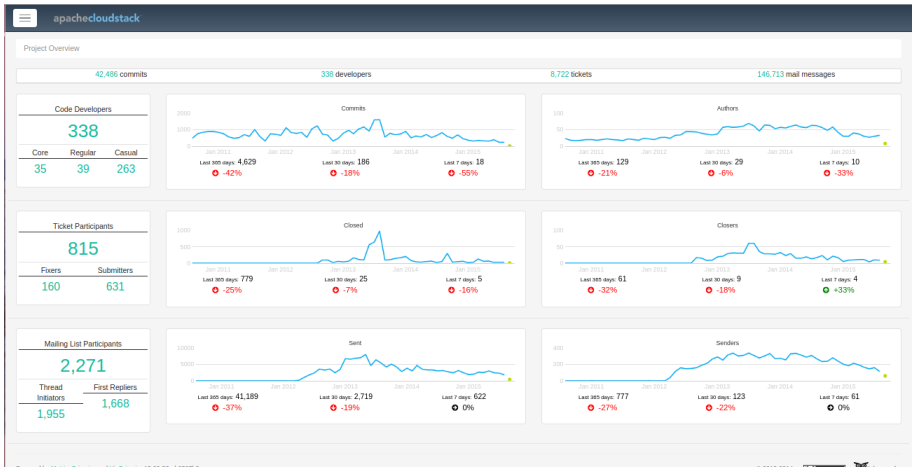

Figure 3.2: Apache Cloudstack Dashboard

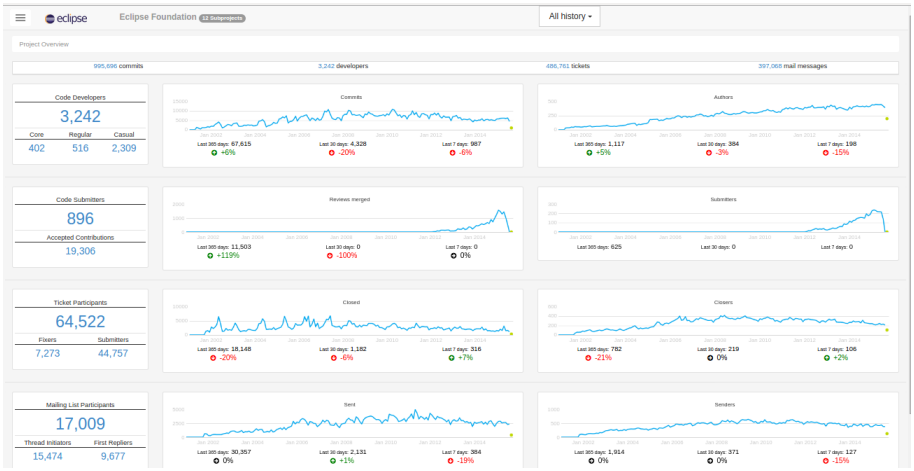

Figure 3.3: Eclipse Dashboard

3.2 Task at hand

We looked at the existing visualizations by Bitergia for the two cases that we picked to work on and found that they were not feasible to answer different kinds of questions that one could pose. We had to work through various visualizations in order to arrive at the first design adaptation. We will first talk about each of the two cases presented to us and then discuss the solutions that we fixed upon in the next chapter.

CASE 1 - Experience of Developers

A lot has been discussed about the motivation for developers to join or stay or leave an open source project. The most sort after work being the onion model which describes the general roles that developers take and how they gravitate eventually [11, 6]. The Bitergia team focused on calculating age in project for

active developers at a certain time. They had access to datasets which gave them the data needed to calculate the same. As a result of their work the following visualizations were built -

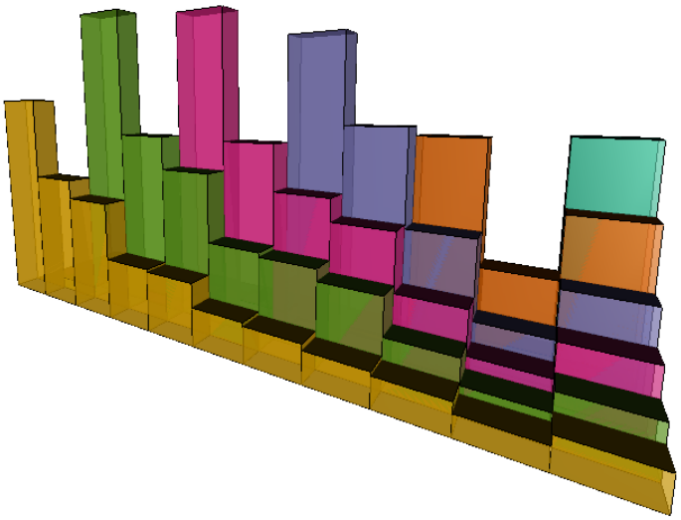

Figure 3.4: 3D version of pyramids every two years

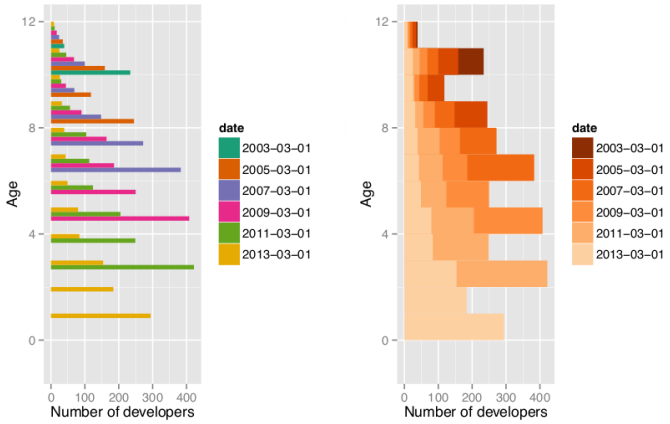

Figure 3.5: Comparison for pyramids every two years

The problem here was that, the 3D visualization, though colorful, was a little

hard to relate to, to understand the dynamics. Also the bar charts were an absolute mark of the developer numbers at any given time but they were not enough to know which group of developers were actually staying and which ones were leaving. Turnover and age were not quite well depicted was the intial thoughts of the team.

CASE 2 -Where do Developers work ?

Location has always been of importance in any area of work. In the software industry, distance always was not a matter of concern because of the nature of work involved, with this being even more true for Open Source Software as its developers are assumed to be contributing according to their convenience [5]. Considering this, the group at Bitergia decided to investigate the overall spread of developers across the world for a few projects. Their main database was the data they collected from github and mailing lists for those organizations. There were certain assumptions involved like the timezones mentioned in the mail tools were correct and that they correspond to the geographical areas to some extent. They came up with these two basic visualizations as a start to analyzing the data.

The following was the division of geography with respect to timezone -

1. America: GMT-8 to GMT-2 (US/Canada: -8 to -4)
2. Europe/Africa/Middle East: GMT to GMT+5
3. East Asia/Australia: GMT+8 to GMT+11

The visualizations being very basic didn't help solve or understand a lot about the developer spread for an organization. They were more oriented in giving an overview of messages per timezone. It was not clear about the relation between commits and authors and location tied together which needed more than a birds'

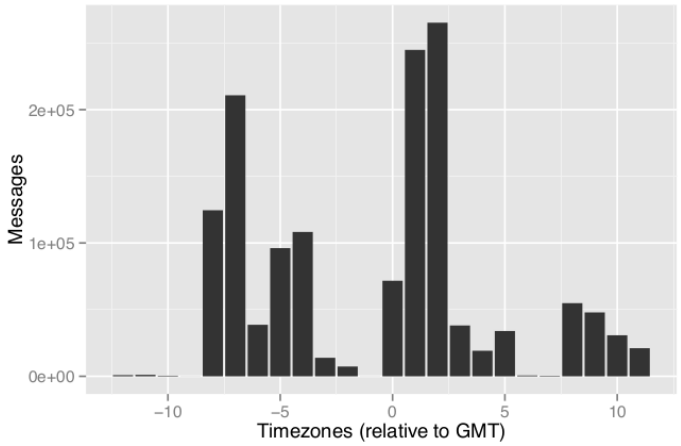

Figure 3.6: Timezone origin of messages

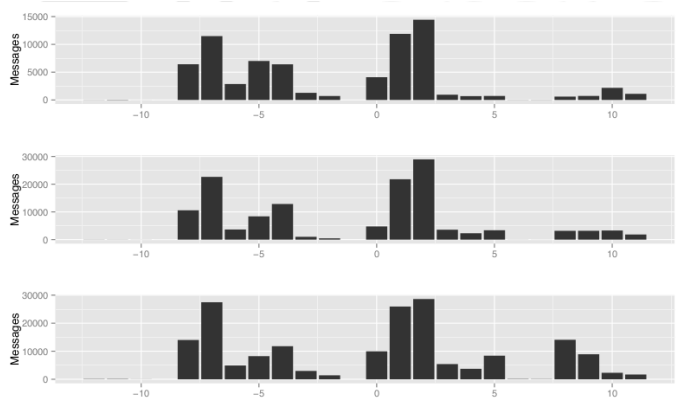

Figure 3.7: Timezone origin of messages for 2002,2007,2012

eye view type of a visualization.

Chapter 4: Methodology

To help explore the previously explained cases, our task was to look into improvising the visualizations to help people understand the data in a better way. To carryout this process we took datasets of 5 open source projects for the cases explained in the previous chapter. We started off with different prototypes for each case and improvised by taking timely suggestions from the team. We build initial visualizations and followed an iterative process of development.

4.1 The Organizations

1. Eclipse Foundation : This is a member supported open source community focused on building a platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing softwares across the lifecycle [36].

2. OpenStack : It is one of the most active open source products currently available for all types of cloud environments. It controls a a large number of resources for computation, storage ,networking for a datacenter [37].

3. Puppet Labs : This is an open source tool which manages various stages of IT infrastructure inclusive of provisioning, patching, configuration and management of operating systems across cloud structures [38].

4. RedHat RDO: This is a community of people who are interested to use and deploy OpenStack on the RedHat Linux, Fedora ditributions. It is a sub community of people who use RedHat which is one of the largest contributors to Linux [39].

5. Citrix Apache Cloudstack : This is an open source software which deploys and manages networks of virtual machines as a IaaS(Infrastructure as a Service) platform. It supports VMware, KVM, XenServer, Hyper-V [40].

4.2 Why the 5 projects

The choice of projects was done mainly keeping in mind that we cover a variety of projects. In the selected 5 projects,Eclipse Foundation and Red hat have been around and successful for a long time garnering attention from time to time. These projects also have a large developer community and can hence be classified as a large scale project. OpenStack Foundation,Puppet Labs and Apache Cloudstack are relatively newer projects gaining popularity slowly amomg the community. OpenStack has made its presence felt strongly as cloud based software is now gathering a lot of buzz. Citrix's Apache Cloudstack was set up as a result of the gaining popularity of OpenStack, to help people using it on Linux. This gives us a set of medium to small sized projects with growing communities.

4.3 Dataset

As Bitergia already had a large database for a few open source projects, they were very co-operative and gave us full access to all the data that they had in hand for the 5 organizations we were looking into. The files were all in the json format and this was an ideal format for us to work with as well. There was some manipulation of data involved for a few visualizations. The first step was to identify the format of the data and then brainstorm for ideal visualizations depending on the possible ways to understand the data. All the data collected was for February, 2015 and has data from 2010 to 2015. The timezone files have data from timezone -12 to +11.One important field added to the timezone files, was the approximate population for each of the timezones which was taken from the below graph.

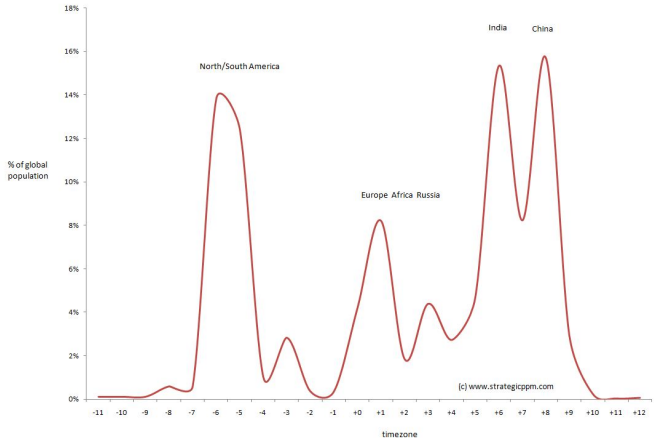

Figure 4.1: Global population by timezone

Also in order to build a timezone based visualization, a geojson file was needed which would draw the map, timezone wise and this file was made available by a

researcher on Techslides. As geojson files have large overhead and would take a lot of time to load, converting it to topojson format was recommended [23]. Topojson is a command line application which not only reduces the size of geojson files, it also gives multiple options like simplification,conversion etc.It can be installed easily using NPM (node package manager).

For developer experience section, we had json files with joining date and last visible action date for the developers of the 5 organizations. The task was then to categorise the developers into different sections to find how many of them stayed for what intervals of time. A pattern was then searched for, to see if the joining and leaving rates were consistent or showed any anomalies. The json files were modified to find these generation ranges and a csv file with this data was created.

4.4 D3.js

As the visualizations needed flexibility the most ideal tool in hand was D3.js. D3 is a javascript library built by Mike Bostock which has the capacity to build interactive, dynamic and a big variety of visualizations. It is largely based on HTML5, CSS and SVG. It makes rendering the visualizations on any browser extremely convenient and easy as well. Due to the large developer support available for the library, it was an ideal choice to build the final visualizations in. It accepts external data in the form of json,CSV or TSV. Being a javascript library, it follows the same syntax hence making data manipulation quick. D3 API has hundreds of functions which make gives it its community. This library is widely used by big

corporates like Datameer, The New York Times, OpenStreetMap etc.

4.5 Experimental Designs

4.5.1 Bitergia's Implementations

We first go onto view the implemetations done by team Bitergia to analyse the data set in hand. Figure 4.2 is used to show the experience of developers and makes use of a bar chart. The y axis is age in quarters for long a developer has stayed and the x axis is the number of developers.

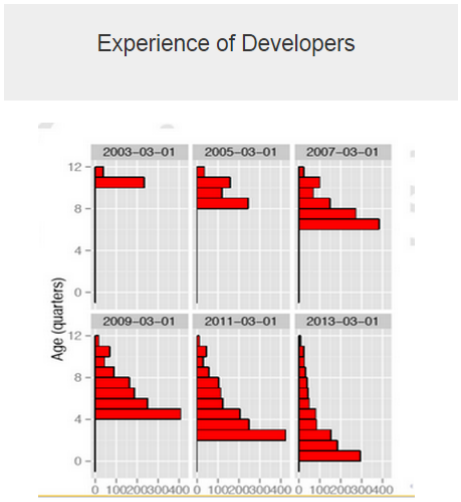

Figure 4.2: Experience of Developers

Figure 4.3 is used to view the number of messages incoming from different timezones. The y axis is number of messages and the x axis is the timezones of

the world. These two implementations were created out of the same datasets that we used for our implementations as well.

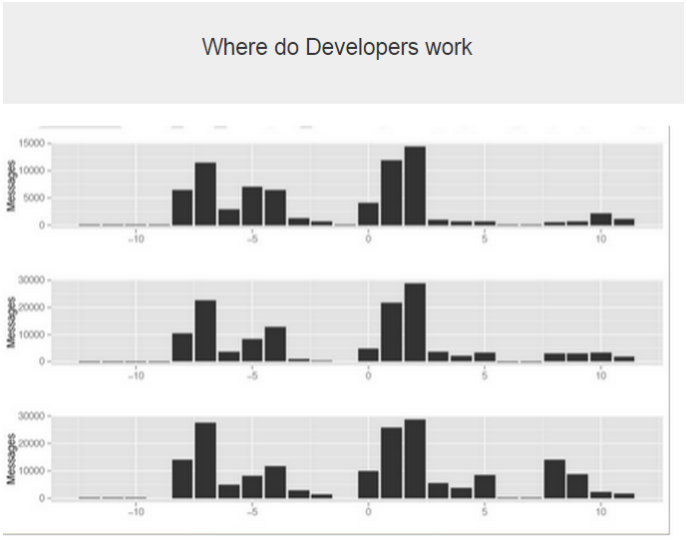

Figure 4.3: Where do Developers Work?

4.5.2 Our Implementations

Figure 4.4 is a bar grouped bar chart which has two main data sets - one column shows the number of developers who join the project each year i.e New developers and the second column shows the number of developers that are still a part of the project each year i.e Established developers.

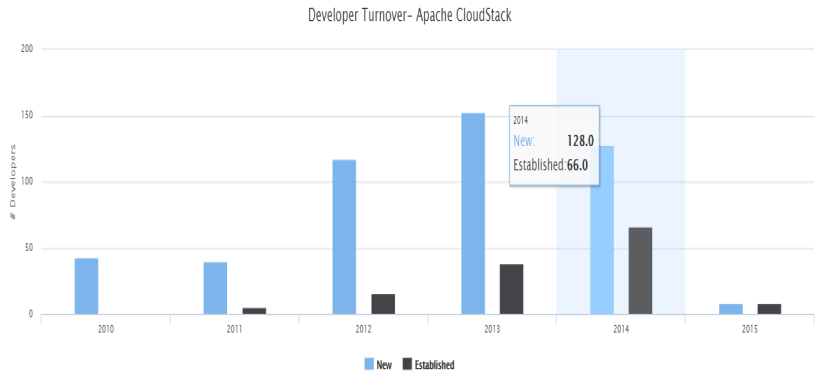

Figure 4.4: Developer Turnover

Figure 4.5 is a stacked bar chart which gives an individual a total look at the number of developers that are still active in a project over the years.

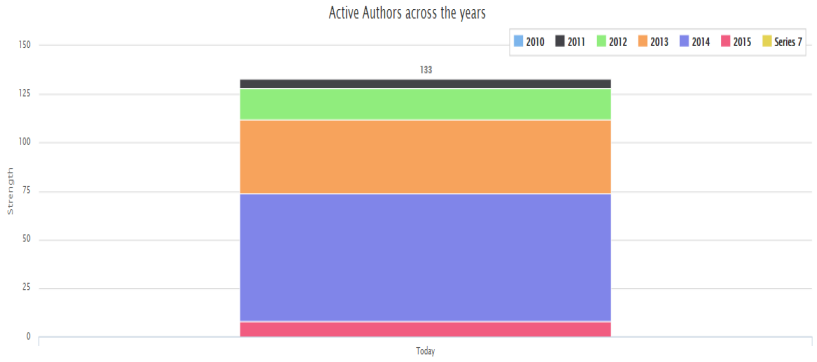

Figure 4.5: Active Authors Across the Years

Figure 4.6 is a also a stacked chart which shows the conversion rate i.e percentage of developers that stay back in a project each year and the mortality rate i.e percentage of developers that leave the project each year.

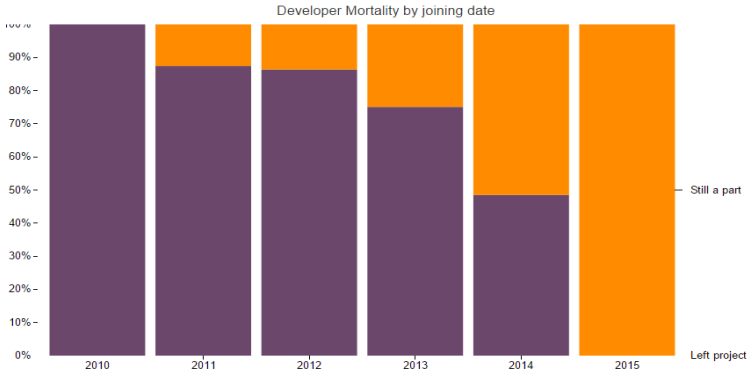

Figure 4.6: Developer Mortality by Joining Date

From Figure 4.7 to Figure 4.13 are all map based visualizations which use color as a main encoding mechanism to show one parameter from the dataset. Figure 4.7 is shaded as per number of Authors present in each timezone and the data picked from github. Similarly Figure 4.8 is shaded with the same parameter but the data is taken from the mailing lists.

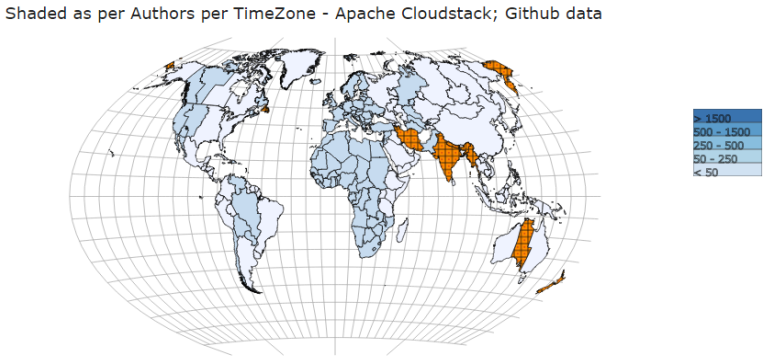

Figure 4.7: Authors per timezone - Github

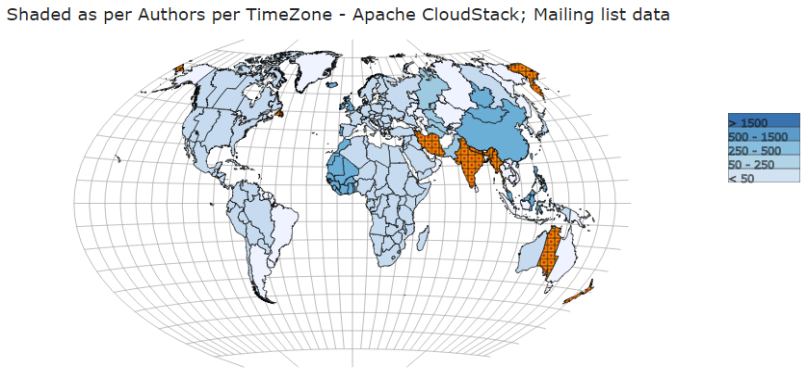

Figure 4.8: Authors per timezone - Mailing lists

Figure 4.9 and Figure 4.10 are shaded as per commits per timezone, data taken from both Github and Mailing lists.

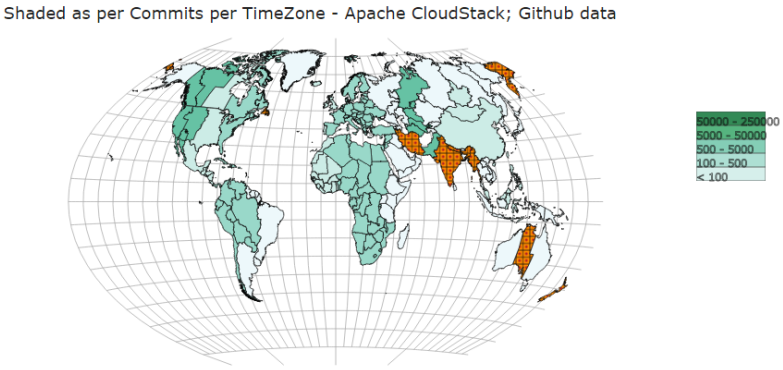

Figure 4.9: Commits per timezone - Github

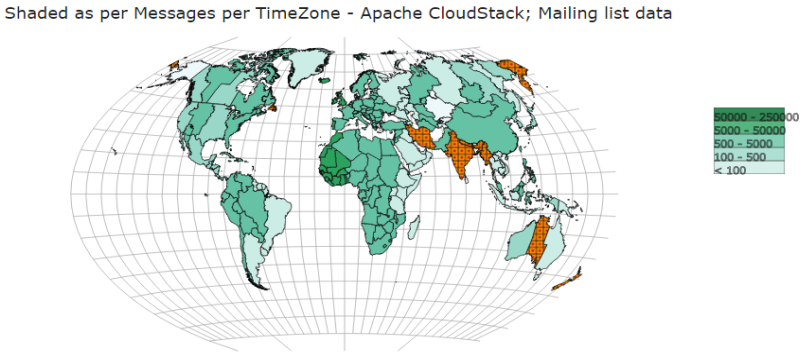

Figure 4.10: Messages per timezone - Mailing lists

Figures 4.11, 4.12 and 4.13 are maps which are shaded with different parameters and the aim here was to try and find a norm or trend that existed for each of the projects.

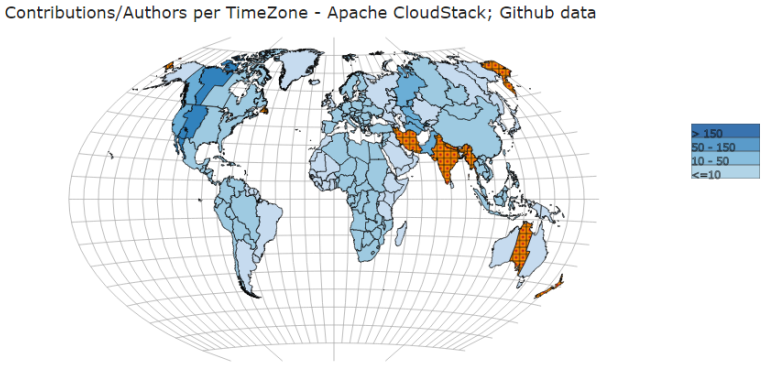

Figure 4.11: Contributions/Author per timezone - Github

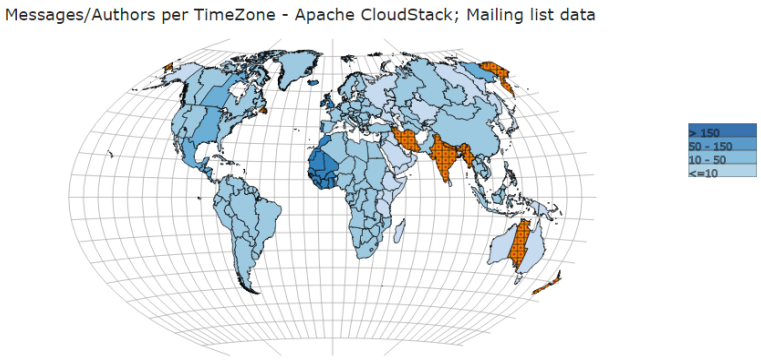

Figure 4.12: Messages/Author per timezone - Mailing lists

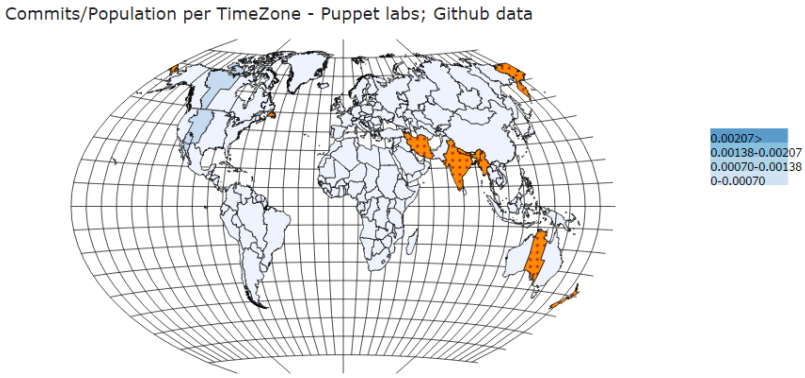

Figure 4.13: Commits/Population per timezone - Github

Chapter 5: Evaluation

In order to gauge the quality and impact of the visualizations, we conducted a focus group study to get a first hand review of the implementations. Focus group sessions are an effective qualitative research method, the main aim of which is to allow the participants to express their point of views and give researchers valuable feedback [21]. Focus groups are used to gather information for discovery, benchmarking, evaluating, verifying perceptions, feelings, opinions and thoughts [22]. Focus groups are most productive when used to determine information on new proposals or solutions, determine the strengths and weaknesses of a solution, assessing whether a solution is working and in the evaluation or success of a concept [24]. Having understood this, we proceeded with the focus study as this was the best fit tool for evaluating our implementation of visualizations.

5.1 Participants

We recruited 6 participants over the age of 18 through the OSU engineering groups by sending out recruitment emails and participant's word of mouth. Table 1 gives an overview of participant demographics. All the participants had some sort of prior experience with software development and have worked with data visualizations at some level. They were made to sign the consent forms before the session

Participant Code Name	Degree and Major	Age	Gender
A1	PhD, Computer Science	25	Female
A2	PhD, Computer Science	27	Male
A3	PhD, Botany and Plant Pathology	30	Male
A4	Masters, Computer Science	25	Female
A5	Masters, Computer Science	24	Male
A6	Masters, Electrical and Computer Engineering	22	Female

Table 5.1: Participant Demographics

started and were also asked for permission to be recorded. Notes were taken and, if audio recorded, transcribed by the researcher.

5.2 Study Session

The participants on arrival were given a nametag with their code names and scenario numbers. Three of the participants were asked to look at the visualizations done by Bitergia and three of them were given the visualizations done by us. The goal was to make the participants use both the visualizations and then have a discussion about their effectiveness. The participants were not told beforehand which visualizations they would be handling first. After having completed the tasks with one scenario, the participants then swapped and completed the same tasks with the second scenario. This was a within group study as all participants performed both tasks. This took up almost 40 minutes of the session. Once done, the participants were then led to a discussion session where they were asked some

questions with regard to the visualizations to bring out their opinions and views about each of the implementations. The questions that were posed basically tried to cover various metrics which would be used to evaluate the visualizations.They were mostly open ended and covered the following topics: a)How close to context were the visualizations, b)were they easy to understand and use, c)were meaningful outcomes being drawn, d)how they dealt with missing data etc. The discussion was facilitated by a researcher and the participants were probed from time to time in case they deviated from the discussion path. All the participants were very enthusiastic and brought out some very interesting points during the course of the discussion. The session lasted for about 2 hours and the collected results were then analysed by means of thematic coding.

5.3 Evaluation Metrics

The evaluation of visualizations has always been subject to a lot of research. The oldest or first ever possible research done in this area was by Tufte in 1983 [25]. Since then lot of researchers have tried their best to formulate a set of guidelines for evaluation yet no concrete measurable metrics are available today to say that yes this visualization is the best. In 2007, Nico Marrero wrote a paper which spoke about possible metrics derived by various researchers, to evalaute the effectiveness and usability of visualizations [31]. These metrics were divided into three main categories;Mathematical Metrics, User Centric Metrics and Effectiveness Metrics. Though not a high standard and highly accepted benchmark, it still has some

credibility when it talks about visualization effcetiveness. From this list we picked 2 metrics under each category which were close to our expectations to define the usability of the visualizations.

Mathematical Metrics

1.Reference Context: A visualization needs to provide a context in order to be comprehendable from most viewpoints. This is hence the number of visible data points in any given visualization with reference to a context [32].

2.Number of data points and data density: This metric was defined by Tufte when he assumed that the more the amount of data made available per square centimeter of screen size, the more effective is the visualization. The number of data points here mean the discrete number of data sets/values present on the screen at any point and data density is the number of points/pixels on the screen excluding the borders and menu options. It is said that visualizations which cannot show atleast 500 data points, are not found to effective [32]. But this is not a fixed lower bound and can vary depending on the use of the visualization. The same goes for the upper bound, there could be visualizations which have around 100,000 data points and be considered effective and there could be visualizations with just 3000 data points and still be deemed as complex [32].

User Centric Metrics

3.Identify Clusters: The ability to group similar data points for better understanding [33].

4.See all features and data: The visualization should be able to show all of its underlying features and data sets without any occlusion [33].

Effectiveness Metrics

5.Dealing with uncertain,missing and dirty data: How effectively can a visualization handle missing and dirty data and bring it across aesthetically to the user [33].

6.Flexibility of visualization: Visualizations need to have the ability to reach the upper bounds of data density or feature listing without becoming too complex to view [33].

5.4 Code Application

Each codeset was applied to the chunked transcriptions of the focus study data. Visualization Effectiveness related codes were applied borrowing methods from grounded theory [34], but using pre-defined codes. Two researchers independently coded all the responses for each scenario and reached a good agreement(Cohens Kappa coeffecient = 0.718 for scenarioA and Cohens Kappa coeffecient = 0.778 for scenarioB).

Chapter 6: Results

In this section we show the results from the tasks undertaken by the participants and then go on to talk about the how the codes were applied to the focus group questions to see how well the scenario's fair when evaluated with regard to the metrics. The questions were all open ended which gave a lot of scope to gain constructive feedback. Here scenarioA refers to our implementations and scenarioB refers to Bitergia's implementation. One thing to be noted is that this was a formative study and was performed to find if we were heading in the right direction and not to gain consensus or state a solution to be better over the other. We were looking at gathering user comments and suggestions for improvement.

6.1 Task Performance

The participants were all given around 3-4 questions in each case for both the scenarios and they had to answer them with the help of the visualizations. Upon completion of the tasks, they told the researcher that they had guessed a lot when it came to answering the questions in scenarioB. They had done it as they could not take the help of those visualizations to answer the questions completely. This raises questions on the validity of their answers and value of their insights. Below is the summarized result of the solutions given to each question for both scenarios.

Find complete responses in the Appendix for these questions.

6.1.1 Case 1 response summary

1. How does the general trend move towards the number of developers joining each year?

   This question was answered well by using the visualizations from both scenarios by all the participants. The barcharts in both scenarios gave a good idea to find the trends of developers joining each year.

2. Is the number of developers joining a project proportional to the number of developers leaving a project?

   This was one question which had ambiguity when scenarioB visualizations were looked at. The participants were not able to clearly demarcate the trend of joining developers vs the trend of leaving developers. However they were all able to answer the questions correctly with scenarioA.

3. Do you notice a trend between the conversion and mortality of developers in the open source projects shown here?

   They could not give a concrete answer here as they had to understand a lot more than what was visualized. These answers were all hazy and unclear in scenarioB. ScenarioA was easy because the there were multiple visualizations and that helped them answer the question.

6.1.2 Case 2 response summary

1. In any given time zone, does a high number of commits indicate a large number of contributor/author presence?

   For scenarioB, three participants said they could not give an answer to the question with the help of the visualization. Three others gave an answer but were not correct. In scenarioA, all of them gave correct answers.

2. Can we sense other trends to attribute to the performance levels at any given time zone like lack of Internet, education access etc. from the knowledge that we possess?

   For scenarioB, one participant who had knowledge of what countries fell under what timezone tried to give the solution. The other 5 said they could not answer due to lack of this knowledge. For scenarioA, all of them answered the question but one of them agreed to have misunderstood the question and gave the wrong answer.

3. Do highly populated regions/time zones have higher contribution rates?

   This question needed knowledge of both countries in a timezone and their populations. Both of which was lacking in scenarioB and needed personal knowledge. They answered it better using scenarioA though not completely accurately.

4. Do highly populated regions/time zones have higher contributor rates?

This question needed knowledge of both countries in a timezone and their populations. Both of which was lacking in scenarioB and needed personal knowledge. They answered it better using scenarioA as both country data and population in the form of shading was shown.

6.2 Focus Group Data Analysis

1. **Were you able to answer all the questions in both scenarios with the help of the visualizations?**

   All the participants found scenario B to be a little low on information. They found scenarioA to be easier to grasp and make sense of in most cases. They found the case1 i.e age of developers relatively easy to interpret in both scenarios but the case2 i.e where do developers work part was difficult in scenarioB as they weren't sure what to make of the number -10,-5,0,5,10 in the x axis. Most participants agreed to have guessed a lot of the asnwers for the scenarioB questions as they had a tough time analysing and tying back the questions to the visualizations. As part of the coding, both reserachers found that the participants felt a high reference of context with regard to the visualizations in scenarioA with 5 of them making almost direct comments with this regard.

   Reference to Context

   "ScenarioA was much better atleast to interpret.I was able to find all that I needed from it."- A5

2. **Were the visualizations confusing to understand or seem impractical at any point?**

   The participants had one major issue here, they found the axis on both visualizations in scenarioB hard to decipher. Also the case1 visualization gave absolute numbers but that is in no way an entity to predict future structuring of a project as they wouldnt know who is leaving and who is staying. Also the case2 visualization issue was that they had no idea how to make sense of the timezone data as most of them didnt know which country fell where. They liked how visualization in scenarioA played with the colors and was intuitive which helped them as they didnt have to tie back the data to the axis all the time. The researchers were found a good mix of reference to context and see all features and data coding metric fit in here with both of them showing up 4 times in the case of scenarioA and once for scenarioB.

   See all the features and data

   "Ah! Confusion no. I felt like the scenarioA, the visualization was a lot more intuitive, the use of colors also helped a lot." - A4

   "I felt like scenarioA was a lot more interactive like you could see little labels popup or something like that so I feel like that gives you a lot more information than just the visualization itself." - A2

3. **How effectively did the visualizations handle missing datasets? Were you able to find the missing/dirty data points easily?**

   The one thing which came out unanimously was the fact that the orange

and dot pattern stood and that helped them know if there was missing/ unavailable data. One participant went on to further explain that in scenarioB there was no way one can know if a certain country's data is not available and countries with half timezones are assumed to be clumped with other timezone data. But this was easily answerable in scenarioA and it segregated missing data, half timezone data and clean data very well. They did point out that the knowledge of a country's population is personal and it would have been nice to view this on the map somehow as well.All 6 particpants made direct response to the feature of handling dirty data. We also coded see all features and data in a few responses.

Dealing with uncertain, missing or dirty data

"While scenarioA, its very clearly said as you have that orange color with dots saying its missing and thats a good way of saying we don't have data there." - A2

4. **Can we predict future structure of open source projects using the visualizations in either scenario?**

   Developer mortality rate very useful visualization as it showed the joining and leaving numbers very well. Also developer turnover visualization gave insight on how many joined the project which can help gauge the interest in a project. ScenarioA shows dynamics of a project over scenarioB which showed numbers and is not of a lot of interest to them.One participant felt that both scenarios had something to offer with A being on the higher end but

if they were to be combined and viewed that would add a lot of interesting facts maybe. Identifying data clusters was a high preference code with 5 participants reply close to it. Reference to context was a close second place with a frequency of 4 codes.

5. **Describe the effectiveness of both scenarios in attaining a meaningful outcome as a researcher**

   They felt a useful visualization should be able to handle the data very efficiently and use metrics correctly to show multiple data points, both of which were highly available in scenarioA visualizations. The fact that there were multiple graphs in scenarioA as compared to scenarioB pointed out a lot of things to them as different data points were all brought out together and it showed beautiful trends. Again a participant added, a visualization can only be effective when it can serve the purpose of answering the question being asked. Data density, reference to context and see all features and data were the codes applied to most repsonses with each of them appearing 4 times.

   Data Density

   "It is always important to think about user space whenever displayin visualizations, how does it fit together, how all the data shows up and fits together in that space, how the colors interact with each other and I think the scenarioA did a lot better there and did a very good job." - A3

6. **Would adding more data points clutter the visualizations or add to its flexibility?**

Most participants felt that the map visualizations had a lot of potential to show a lot more data by making use of different encodings. One suggested use of colors for one dataset and thickness of border lines for another dataset. Another suggested the use of layering on the maps and clickable countries to add another visualization with more specific data. One participant felt the traditional approach of using barcharts was of less/no use when it came to showing geographic data and the maps worked in this case for them. Most responses spoke about the flexible nature of the visualizations with a frequency of 4.

Flexibility of visualizations

"You can use colors for commit frequency and other metric you can show with thickness like internet bandwidth in countries, so nothing overlaps, the map visualizations can handle this well." - A1

6.3 Metric Frequency for each Scenario

Metric	ScenarioA	ScenarioB
Reference to Context	23	5
See all features and data	14	1
Flexibility of visualization	6	0
Dealing with dirty, missing and incorrect data	8	0
Data density	3	0
Identify Clusters	15	2
No code applied	9	42

Table 6.1: Evaluation Metric Frequency for each Scenario

The table 6.1 gives an insight on how both the visualizations stood when it came to evaluating them with the the metrics. ScenarioA was highly appreciated for its ease of use, contextual nature and flexibility, while scenarioB had one major issue of being too stagnant and low on information. The visualizations gave little but were not deemed to be completely useless. The participants were of the opinion that if the scenarioB visualizations would be combined with scenarioA visualizations then they might probably add some value to a resarcher's quest.

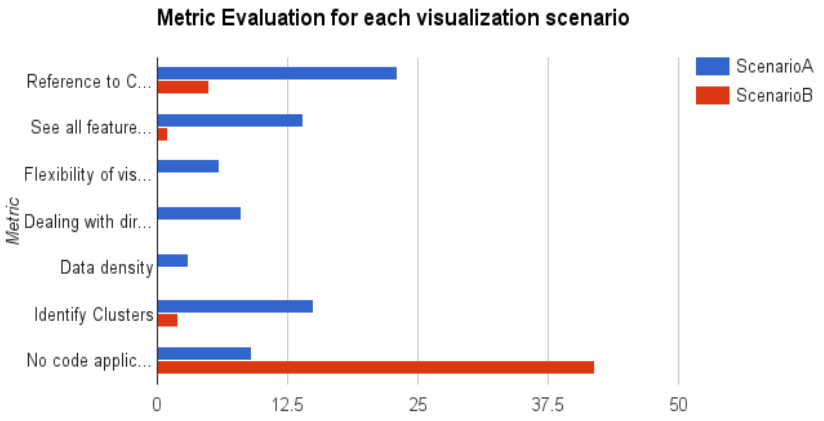

Figure 6.1: Metric Evaluation

Chapter 7: Threats to Validity

7.1 Internal Threats

1. In our study, all participants had to answer the same set of questions for both scenarioA and scenarioB. This creates a confound effect as it is possible that one scenario was actually a lot more effective to answer the questions over the other. So this could lead to a set of participants using the previous scenario's knowledge to answer for the new scenario. We tried to reduce this effect by alternating the order of scenarios given to the participants.

7.2 External Threats

1. The participants of the study might not be representative of the actual target population.

2. The tasks chosen might not be completely representative of the tasks looked at by researchers.

7.3 Reliability Threats

1. As participants might not be reprsentative of the target population, the results/findings might not be replicable in real world.

2. In cases where the participants guessed the answers for either scenario would lead to reliability issues.

3. As the study was conducted by the researcher alone the possibility of missing out on a few notes/comments from the participants or mis interpretation of the data is possible.

Chapter 8: Summary

A large number of developers contribute to open source projects from all over the world and it becomes important to understand their working behaviour to gauge their interest in the project and the growth of the project.Visualizations add a lot of drama to a dataset making it reflect things that the naked eye can't always see. In our case the aim was to make visualizations which would help understand the way developers work/contribute to an open source project to infer the future scope of these projects. As stated by one of the user study participants, "It all depends on what the message is and how it can be conveyed effectively". Both visualizations had their own shares of advantages and disadvantages, but scenarioA stood out plainly because it gave out the message clearly and in multiple ways. ScenarioB was not thrown out by our participants, they felt it had potential too, if modified and looked at, along with the visualizations in ScenarioA. Having said this there is room for improvement in our imlplementations as well as we got a wide variety of suggestions from the participants and also due to the small size of the group there is also scope for conducting further studies to get more feedback.

Bibliography

[1] Robles G , Barahona M Jesus. Geographic Location of Developers at SourceForge, *MSR*,2006.

[2] The Open Source Definition - Annotated. http://opensource.org/osd-annotated.

[3] Ten Reasons Open Sourceis Good for Business. http://pcworld.com/article/209891

[4] David Tuma Open SOurce Software: Opportunities and Challenges In *CrossTalk- Journal of Defense Software Engineering*, 2005.

[5] Takhteyev Y , Hilts A. Investigating the Geography of Open Source Software through Github, 2010.

[6] Ye Y , Kishida K. Toward an Understanding of the motivation of Open Source Developers. In *ICSE*,2003.

[7] Sethanandha B.D , Massey B, Jones W. Managing open source contributions for software project sustainability, In *PICMET*, 2010.

[8] Linus Torvalds quoted in The Cathedral and The Bazaar

[9] Madey G, Freeh V, Tynan R. The Open Source Software Development Phenomenon: An analysis based on social network theory, In *8th Americas Conference on Information Systems*, 2002.

[10] Brenda Chawner Community Matters Most: Factors that Affect Participant Satisfaction with Free/Libre and Open Source Software Projects In *ACM*, 2012.

[11] Crowston K , Howison J. The Social Structure of Free and Open Source Software Development. In *First Monday*, 2005.

[12] Von Engelhardt S, Freytag A, Schulz C. On the Geographic Allocation of Open Source Software Activities. In *Jena Economic Research Papers*, 2010.

[13] Jesus M Gonzalez Barahona Measure your open source communitys age to keep it healthy. http://radar.oreilly.com/2014/10/measure-your-open-source-communitys-age-to-keep-it-healthy.html

[14] Ghosh R A, Glott R, Krieger B and Robles G. Free/libre and open source software: Survey and study, In *Report,International Institute of Infonomics,University of Maastricht, Maastricht, The Netherlands*, 2005.

[15] Tuomi I. Evolution of Linux credits file: Methodological challenges and reference data for open source research, In *First Monday*, 2004.

[16] Igor Steinmacher, Marco A. Graciotto Silva, Marco A Gerosa Barriers faced by newcomers to open source projects IN *OSS*, 2014.

[17] David P, Watermann A, Arora S. FLOSS-US: The Free/Libre/Open Source Software Survey for 2003, In *Stanford Institute for Economic and Policy Research*, 2003.

[18] Mockus A, Fielding R, Herbsleb J. Two case studies of open source software development:Apache and Mozilla, In *TOSEM*, 2002.

[19] https://en.wikipedia.org/wiki/Anscombe

[20] Linux Foundation Survey https://linuxfoundation.org/publications/linux-foundation/.

[21] Judith A. Villard. Use of Focus Groups: An Effective Tool For Involving People in Measuring Quality and Impact.

[22] Patton M Q. Qualitative Evaluation and Research Methods.

[23] http://techslides.com/time-zone-world-map-with-d3-and-topojson.

[24] Greenbaum T L. The handbook for focus group research.

[25] Tufte E R. The Visual Display of Quantitative Information, In *Graphics Press, Cheshire, CT*, 1983.

[26] Ghosh R, Krieger B, Glott R, Robles G. Free/Libre and Open Source Software: Survey and Study, *International Institute of Infonomics, University of Maastricht*, 2002.

[27] Syeed M, Hammouda I, Systa T. Evolution of Open Source Software Projects: A Systematic Literature Review, In *Journal of Software*, 2013.

[28] Oriol M, Oscar F D, Franch X, Marco J. Assessing Open Source Communities' Health using Service Oriented Computing Concepts In *IEEE RCIS*, 2014.

[29] E. Ververs, R. van Bommel, and S. Jansen. Influences on developer participation in the debian software ecosystem, In *International Conference on Management of Emergent Digital EcoSystems (MEDES), ACM*, 2011.

[30] Goeminne M, Mens T. A framework for analysing and visualizing open source software ecosystems, In *13th International Workshop on Principles on Software Evolution, ACM*, 2010.

[31] Nico Marrero. Visualization Metrics: An Overview.

[32] Richard Brath. Metrics for Effective Information Visualization, In *IEEE Symposium on Information Visualization*, 1997.

[33] Grinstein, G., Hoffman, P., Laskowski, S., Pickett, R. Benchmark development for the evaluation of visualization for data mining, In *Information Visualization in Data Mining and Knowledge Discoverey*, 2001.

[34] J. Corbin, A. Strauss. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, In *SAGE*, 2008.

[35] Bitergia. https://www.bitergia.com/

[36] Eclipse Foundation. https://www.eclipse.org/

[37] Openstack Foundation. https://www.openstack.org/foundation/

[38] Puppet Labs. https://puppetlabs.com/

[39] Redhat RDO. https://www.rdoproject.org/

[40] Apache Cloudstack. https://cloudstack.apache.org/

APPENDICES

Appendix A: Task Responses

ScenarioA responses

Case1

1.How does the general trend move towards the number of developers joining each year?

A1	Apache CloudStack: The project probably started in 2010 (no established developers), and with a sizeable community (40-50 people) . Perhaps it was outsourced then. And then, there was an increase in the number of developers joining between 2010-2013. Decreased in 2014 and dropped in 2015. The ratio of new to established developers also followed a similar trend until 2012, and dropped between 2013-2015.
A2	It grows every year with a taper in 2015. Probably because the year isn't over yet.
A3	It was increasing, but now it's static
A4	There has been an increase in developers joining the project till 2013, but the numbers start to decline starting 2014:Apache Cloudstack
A5	The trend moves on pretty slow, and slowly increases until it reaches maturity period. After that either new technology may take over the existing or the existing technology would be stagnated at certain point
A6	In most cases until 2014 it increased but there was decrease in number of developers in 2015.

Table A.1: Task Response 1 - scenarioA, case1

2.Is the number of developers joining a project proportional to the number of developers leaving a project?

A1	Yes
A2	No. The number of developers leaving does decrease over time, while the number of incoming developers increases
A3	No. There appear to be more leaving the project after a few years
A4	They are somewhat proportional. The number of people that decide to stay are very low compared to the number of people that joined.
A5	No, the number of developers joining the project is not proportional to once who are leaving. It actually depends on the type of project, popularity and whether it satisfies the interest of the ever-changing demands
A6	Yes

Table A.2: Task Response 2 - scenarioA, case1

3. Do you notice a trend between the conversion and mortality of developers in the open source projects shown here?

A1	Both rise, until in 2014 it stabilizes. More people stayed, and less newer people joined.
A2	The conversion rate seems to increase while the mortality rate decreases.
A3	Yes. As the project grows, more of the early developers are gone.
A4	The conversion rate seems pretty low compared to the mortality rate.
A5	Yes
A6	In most cases mortality rate decreases

Table A.3: Task Response 3 - scenarioA, case1

Case2

1.In any given time zone, does a high number of commits indicate a large number of contributor/author presence?

A1	No.
A2	Yes. I can see the number of authors and the number of commits, so I can see a correlation between the two.
A3	Yes
A4	It depends on the time that people prefer to make their contributions. The evening for one country might be the morning time for another country and people might be making contributions in both countries because come people might be making contributions at work and other might be making contributions after work.
A5	Yes
A6	Yes.

Table A.4: Task Response 1 - scenarioA, case2

2.Can we sense other trends to attribute to the performance levels at any given time zone like lack of Internet, education access etc. from the knowledge that we possess?

A1	No, Africa for example, has a high number commits per timezone.
A2	Timezones clump the world longitudinally. Most of Africa is on the same time one as Western Europe. Same applies for South America with North America. The only place when such a trend can be seen is Asia, where one would expect a higher number of contributors due to a higher population.
A3	No
A4	The lack of internet and education are very big factors. Even if you have a large population, and the population is not very educated then the contribution rates are going to be low.
A5	No.
A6	Yes.

Table A.5: Task Response 2 - scenarioA, case2

3.Do highly populated regions/time zones have higher contribution rates?

A1	No, Africa for example, has a high number commits per timezone.
A2	Timezones clump the world longitudinally. Most of Africa is on the same time one as Western Europe. Same applies for South America with North America. The only place when such a trend can be seen is Asia, where one would expect a higher number of contributors due to a higher population.
A3	Yes.
A4	The lack of internet and education are very big factors. Even if you have a large population, and the population is not very educated then the contribution rates are going to be low.
A5	No.
A6	Yes.

Table A.6: Task Response 3 - scenarioA, case2

4.Do highly populated regions/time zones have higher contributor numbers?

A1	No, again South Asia or China does not show that trend.
A2	No. Same answer as previous.
A3	Yes.
A4	No.
A5	May be, again it depend on quality of education,awareness, technology affecting the people over there.
A6	Yes.

Table A.7: Task Response 4 - scenarioA, case2

ScenarioB responses

Case1

1.How does the general trend move towards the number of developers joining each year?

A1	Increases. However, there is an increase in the number of developers with lesser experience.
A2	It grows. The labels on the X axis are hard to read
A3	Yes.
A4	More developers are joining later in the year.
A5	It is gradual and slow, and awareness about type of technology which interest may interest developer and organisation.
A6	10.

Table A.8: Task Response 1 - scenarioB, case1

2.Is the number of developers joining a project proportional to the number of developers leaving a project?

A1	Yes, the leaving happens in the top part of the graphs (higher experience) and the joining happens in the lower part. Also, these are regular triangles, with almost similar area. Hence the number of developers seems to remain the same, while the average experience shifts to the bottom, year on year.
A2	I cannot say. I only have absolute numbers, and the visualization does not show people leaving or entering the project.
A3	This is not clear. I'll say.... No?
A4	In my opinion the number of people joining a project is greater than the number of people leaving the project. Because of this the number of people at any time working on a project has increased consistently over the years.
A5	No, it depends on the type of technology, interest of an organisation and developer. More likely developers interest plays an important role I feel.
A6	In most scenarios it is proportional..

Table A.9: Task Response 2 - scenarioB, case1

3.Do you notice a trend between the conversion and mortality of developers in the open source projects shown here?

A1	Conversion happened in the developers with lesser experience, and increasing number of more experienced developers left.
A2	No
A3	Not particularly.
A4	Not sure.
A5	–
A6	Yes, in general they tend to increase.

Table A.10: Task Response 3 - scenarioB, case1

CAse2

1.In any given time zone, does a high number of commits indicate a large number of contributor/author presence?

A1	Does not show it.
A2	No really. It could be just one very active developer in that region
A3	Assuming x is time zone, I don't have enough information since y is messages.
A4	Yes.
A5	Yes.
A6	Yes.

Table A.11: Task Response 1 - scenarioB, case2

2.Can we sense other trends to attribute to the performance levels at any given time zone like lack of Internet, education access etc. from the knowledge that we

possess?

A1	Does not show it.
A2	Well, most commits are from the UTC to UTC-2 which contains western Europe and around UTC-5, which contains North America. Some of the time zones contain only water, so it's natural that no developers live there. Also, good parts of Africa are also under the same time zone as Europe. So it is very difficult to distinguish between these two time zones. Also, the same applies for South America
A3	No.
A4	Some people may not have internet access in their houses so all their contribution may happen at work, but it does depend on the country that the developer is from.
A5	Yes.
A6	yes several other factors tend to influence the performance levels.

Table A.12: Task Response 2 - scenarioB, case2

3.Do highly populated regions/time zones have higher contribution rates?

A1	Does not show it.
A2	No. UTC+10 and around does contain India and China, which are heavily populated. However, the number of messages does not spike there.
A3	Again, assuming x is time zone, I don't have enough data.
A4	Just because the population is high does not mean that the everyone in that country have the knowledge required to make contributions.
A5	Depends on the knowledge level, and technology trend in that time zone. People would contribute to that technology which is more likely used in that time zone.
A6	Yes.

Table A.13: Task Response 3 - scenarioB, case2

4.Do highly populated regions/time zones have higher contributor numbers?

A1	Does not show it.
A2	Again no. Same answer as above.
A3	I don't have enough data to answer that question.
A4	My answer is similar to the previous one.
A5	Again Depends, on the knowledge level in that timezone.
A6	Yes.

Table A.14: Task Response 4 - scenarioB, case2