

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: _____

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.

©Copyright by Rithika Kiran Naik
September 17, 2015
All Rights Reserved

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 2016

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 Kathy 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 Jennifer Davidson, 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 Master's 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

| | <u>Page</u> |
|--|-------------|
| 1 Introduction | 1 |
| 1.1 Why Data Visualizations? | 4 |
| 2 Related Work | 5 |
| 2.1 Experience of Developers | 7 |
| 2.2 Geographic Diversity of Developers | 8 |
| 3 Data Gathering and Processing | 10 |
| 3.1 Services provided | 10 |
| 3.2 Task at hand | 12 |
| 4 Methodology | 16 |
| 4.1 The Organizations | 16 |
| 4.2 Why the 5 projects? | 17 |
| 4.3 Dataset | 17 |
| 4.4 D3.js | 19 |
| 4.5 Experimental Designs | 19 |
| 4.5.1 Bitergia's Implementations | 19 |
| 4.5.2 Our Implementations | 21 |
| 5 Evaluation | 28 |
| 5.1 Participants | 28 |
| 5.2 Study Session | 29 |
| 5.3 Evaluation Metrics | 30 |
| 5.3.1 Mathematical Metrics | 31 |
| 5.3.2 User Centric Metrics | 32 |
| 5.3.3 Effectiveness Metrics | 32 |
| 5.4 Code Application | 32 |
| 6 Results | 34 |
| 6.1 Task Performance | 34 |

TABLE OF CONTENTS (Continued)

| | <u>Page</u> |
|--|-------------|
| 6.1.1 Case 1 response summary | 35 |
| 6.1.2 Case 2 response summary | 36 |
| 6.2 Focus Group Data Analysis | 37 |
| 6.3 Metric Frequency for each Scenario | 41 |
| | |
| 7 Threats to Validity | 43 |
| 7.1 Internal Threats | 43 |
| 7.2 External Threats | 43 |
| 7.3 Reliability Threats | 43 |
| | |
| 8 Summary | 45 |
| | |
| Bibliography | 45 |
| | |
| A Task Responses | 50 |

LIST OF FIGURES

| <u>Figure</u> | <u>Page</u> |
|--|-------------|
| 1.1 Onion model by Ye and Kishida | 3 |
| 3.1 Bitergia data forums | 11 |
| 3.2 Apache Cloudstack Dashboard | 12 |
| 3.3 Eclipse Dashboard | 12 |
| 3.4 3D version of pyramids every two years. The height of each bar depicts the number of developers that stayed for 'x' number of years and each color depicts the data for a particular year. | 13 |
| 3.5 Comparison for pyramids every two years. The colors here point to data from the corresponding years and the height of the bar is the measure of the number of developers in each quarter. | 14 |
| 3.6 Time zone origin of messages | 15 |
| 3.7 Time zone origin of messages for 2002,2007,2012 | 15 |
| 4.1 Global population by time zone | 18 |
| 4.2 Experience of Developers | 20 |
| 4.3 Where do Developers Work? | 21 |
| 4.4 Developer Turnover | 22 |
| 4.5 Active Authors Across the Years | 22 |
| 4.6 Developer Mortality by Joining Date | 23 |
| 4.7 Authors per time zone - Github | 24 |
| 4.8 Authors per time zone - Mailing lists | 24 |
| 4.9 Commits per time zone - Github | 25 |
| 4.10 Messages per time zone - Mailing lists | 25 |
| 4.11 Contributions/Author per time zone - Github | 26 |
| 4.12 Messages/Author per time zone - Mailing lists | 26 |

LIST OF FIGURES (Continued)

| <u>Figure</u> | <u>Page</u> |
|--|-------------|
| 4.13 Commits/Population per time zone - Github | 27 |
| 6.1 Metric Evaluation | 42 |

LIST OF TABLES

| <u>Table</u> | <u>Page</u> |
|---|-------------|
| 5.1 Participant Demographics | 29 |
| 6.1 Evaluation Metric Frequency for each Scenario | 42 |
| A.1 Task Response 1 - scenario A, case1 | 51 |
| A.2 Task Response 2 - scenario A, case1 | 52 |
| A.3 Task Response 3 - scenario A, case1 | 53 |
| A.4 Task Response 1 - scenario A, case2 | 54 |
| A.5 Task Response 2 - scenario A, case2 | 55 |
| A.6 Task Response 3 - scenario A, case2 | 56 |
| A.7 Task Response 4 - scenario A, case2 | 57 |
| A.8 Task Response 1 - scenario B, case1 | 57 |
| A.9 Task Response 2 - scenario B, case1 | 58 |
| A.10 Task Response 3 - scenario B, case1 | 59 |
| A.11 Task Response 1 - scenario B, case2 | 59 |
| A.12 Task Response 2 - scenario B, case2 | 60 |
| A.13 Task Response 3 - scenario B, case2 | 61 |
| A.14 Task Response 4 - scenario B, case2 | 61 |

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(OSS). Open source software is a type of software whose development is volunteer driven [1] and for which the code base is made available for any person to modify, edit and redistribute [2]. Open source software is being heavily relied on and used today because of-

1. Quality: Research has found support for the saying that in Open Source, “Given enough eyeballs, all bugs are shallow” [3]. Furthermore, by being developed by a large and diverse base of developer teams, applications also develop a large variety of features, huge support groups and overall enhanced products [4].
2. Flexibility: Open Source gives users the ability to tweak any part of the functionality to match their needs [4].
3. Auditability: Access to the code base promotes transparency around issues like security, freedom, flexibility etc., something lacking in closed source software [4].
4. Large support base: 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 hold onto a problem long enough.

With a plethora of such important features and advantages, the adoption of Open Source has been steadily increasing.

Managing any software development effort comes with its own set of challenges and OSS is no different. Having mentioned its advantages, managing these distributed communities is a big task. With the code base made available to all, issues like maintaining coding standards, peer review processes, skill sets, documentation etc. are important [5]. The fact that these efforts are volunteer based brings up issues of fairness and equity among the developers. OSS relies heavily on its community for its growth and development [6], making co-ordination among them a foremost point of importance. The OSS communities usually have a lot of developers from all over the world who volunteer to produce code and support the software in general. Their motivations have been open to studies [7]. Unlike traditional software, OSS has a wide spread developer/volunteer group, the success or failure depends strongly on its community, as they can make or break the project. The onion model divides the developer lifecycle into 7 layers, with each layer moving inwards, showing the level of involvement of the developers. The outermost layer is considered as the starting point for a new developer after which they slowly transition to higher points of responsibilities, if they continue to stay with the organization, with the final stage being, becoming a part of the core community and a regular contributor [9].

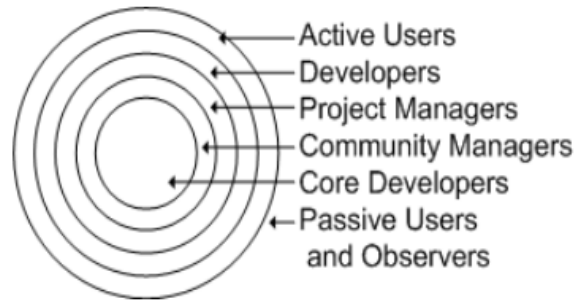


Figure 1.1: Onion model by Ye and Kishida

This being the basis to understand a developer's growth, how can one know the spread of the developers who contribute? Current approaches rely on very biased or incomplete knowledge bases, for instance, finding a large number of developers joining the project and marking that to be as a growing project [10]. As the community benefits by having developers from all across the world, it is important to identify the regions which contribute more to the open source world and the regions which need help. This can help researchers identify why developers in a certain region contribute more or less than others. This was the first motivation for this research. Also, a project's peak phase of success can be noted by looking at the community size and contribution size at any given time. This can then be probably used to predict future structure of the communities for the projects. This became the second motivation.

1.1 Why Data Visualizations?

In 1973, the Anscombe's Quartet showed how four sets with nearly identical statistical properties, appeared very different when graphed [11]. This led to the thought that understanding data by just looking at it might not necessarily be useful or correct. Hence visualizations have gained a lot of importance as tools to promote understanding. As the proverb says, "A picture is worth a 1000 words", visualizations throw light on datasets in a completely new way. As the amount of data increases, finding a common way of unifying the data with its underlying meaning is needed. Comparing, contrasting, reviewing data becomes easier when one views various visualizations. Besides creating visualizations, it is necessary to understand the requirement in hand, perform data manipulation accordingly and then create a suitable visual diagram. Failing to differentiate between excess data and useful data would result in unfavorable results during visualization analysis. The main goal here was therefore to build a neat, easily accessible, intuitive and effective set of visual interfaces which can give an insight on the trends in open source projects with regard to developer's contributions and experiences.

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 continuously 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 reason the Linux Foundation releases a yearly survey to understand how they performed and looks at various variables ranging from code quality, to the size and composition of the developer base, to support issues and their preferred way of showing these varied results are through visualizations [12]. This has become a guideline for most surveys today as it holds a lot of information that would help understand the growth of an open source community. Surveys have been conducted with around 1600+ developers all over the world to look at areas like developer role in open source projects, extent and intensity of open source contributions, support coming in from the core team. The surveys have 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 [13]. But the survey just gave a general overview of developers contributing and made no emphasis beyond that. Another interesting area looked at to understand a developer's motivation was to understand how their roles affected the evolution of the community. Ye and Kishida looked into these motivation factors and stated, "Learning plays a big

part in getting developers to contribute” [9]. This was seconded by the study to understand the development in Apache and Mozilla, where they also found that learning and capability to solve a problem are major motivations for developers at any level in an open source project [14]. A lot of research has been done to understand developer motivations in open source and sustainability factors like number of developers in a project [13, 15], size and quality of the code base [16], number of users for any given project [15], functioning principle of projects [17], etc. Also the evolution of open source projects were 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 [17].

In 2014, research was done to gauge the health of an open source community. This work was done by tools prepared to gauge certain communities or platforms, and were not universal in nature. For example, specific solutions existed for communities like GNOME [18], Debian [19] etc., accessing the data repositories, but the same tools could not be used for other communities [18]. There were frameworks that monitor and analyse repositories that were more generic, but still had their own limitations with technologies [16]. 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 [16].

The most common factor found by researchers while gauging the health and evolution of OSS was the impact the communities had on the projects. It might be

possible that a sustainable community could be attained with a constant inflow of developers with varied skill sets, ages and from a geographically wide area. This might also go onto show a wide spread knowledge about the project and a good mix of ideas and interests.

2.1 Experience of Developers

In 2002, a study was conducted on two open source projects, Apache and Mozilla. 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, in Apache and Mozilla at least, the core group was highly active and apart from them, only a handful of developers, not in the core group contributed on a regular basis [14]. The researchers at Notre Dame University studied the parameter on a larger set of open source projects, and noted that developers in open source projects are not as well connected as developers in other collaborative networks [7]. In an article in 2014 [20], researcher Jesus Barahona stated, “In order to gauge a community’s health it is important to keep track of the developer ages”. He points out that attracting new developers is important, but retaining the experienced ones also was a high priority. The ratio of experienced, long term members to recent ones has a high impact on the quality of code and the ability to support members [20].

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

Age: Measures how long ago each current member joined a project. An indication of how many people are at what stage 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 [20].

2.2 Geographic Diversity of Developers

It is interesting to note that although studies have been conducted to understand the motivations of the developers, understanding the range of developers contributing, geographically or age wise, is still very crude. Greogorio Robles and Gonzalez-Barahona were among the first ones to look into the geographic factors that influenced developer contributions [9]. They collected data of all the registered users from SourceForge, the largest libre software web based platform which had data about geographic locations. They came up with a distribution of developers in different regions of the world. The result of this study showed North America in the forefront with the highest number of contributors, followed by Europe and clusters of Asia, Australia, South America and Africa [21]. The problem here was that the information about location or geography was not captured cleanly in SourceForge. It was mostly left blank during account creation by the account holders. After this came the study by Takhteyev Y, Hilt A and they studied open source data made available by Github, which was becoming a pioneering source

management application. But there were problems here as well as the dataset was either incomplete or was not categorizable by Top Level Domains(TLD) [22]. Then came the idea of looking at time zones and Internet Protocols(IP) of the systems of the users who contribute, both collected through SourceForge [23]. But the issue with looking at IP addresses was that the dataset was not available to all and agreements needed to be signed for the same [23]. Results from the above mentioned studies all show that European and North American countries particularly Unites States of America were the major areas where developers were mostly located and worked from [22, 23]. A survey by Ghosh et al [1] showed the influence of French, American and German developers but the survey that was conducted by Tuomi et al [25] showed varied results leaning a lot towards US, UK, Canada and Germany.

Chapter 3: Data Gathering and Processing

Bitergia was founded by four professors from the Universidad Rey Juan Carlos, Spain in July 2012 [36], trying to further our understanding of the open source community and its growth. The focus was to develop data analytic techniques to mine information about project performance, track developer actions, and areas of improvement and identify risks involved in further development.

I leverage the tools and techniques they developed, in my thesis, adding novel visualizations of the data they generate.

3.1 Services provided

They built tools to extract data from repositories like SVN and Git, from mailing lists, from bug trackers like Jira, Bugzilla, Allura etc. These tools together are known as the *MetricsGrimoire Library* which is accessible to all in Github [43].

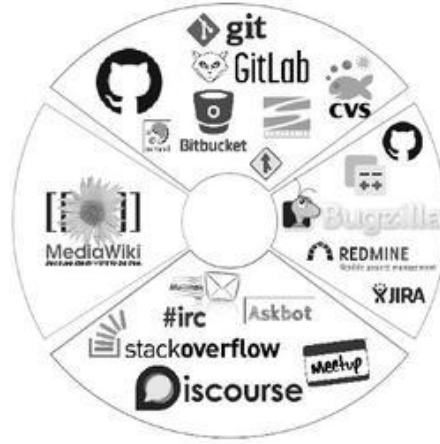


Figure 3.1: Bitergia data forums

With the tools ready to extract data, the next step was to build a framework which would help them analyze and visualize the data obtained from the Metrics-Grimoire tools. This is known as the *VizGrimoire Library*. With such a vast tool base available, it became a lot easier to extract the necessary data for my projects. The dashboards made by Bitergia have been highly acclaimed and used by organizations like Apache, Eclipse, Puppet Labs etc which only added to their credibility.

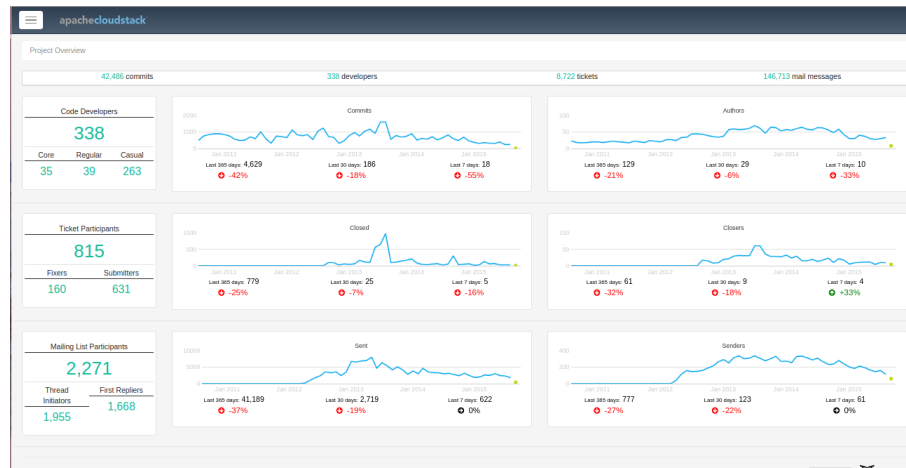


Figure 3.2: Apache Cloudstack Dashboard

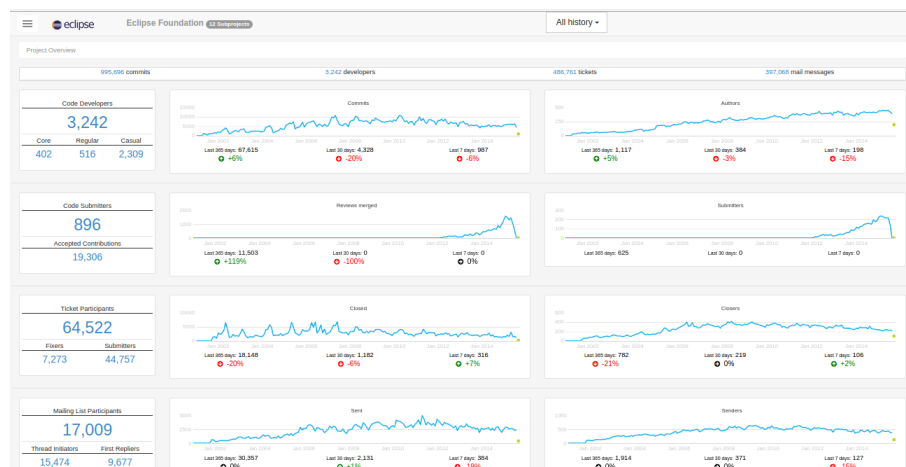


Figure 3.3: Eclipse Dashboard

3.2 Task at hand

Bitergia had some basic implementations done for the two cases previously explored.

CASE 1 - Experience of Developers

The Bitergia team focused on calculating age in project for active developers at a given time. As a result of their work the following visualizations were built-

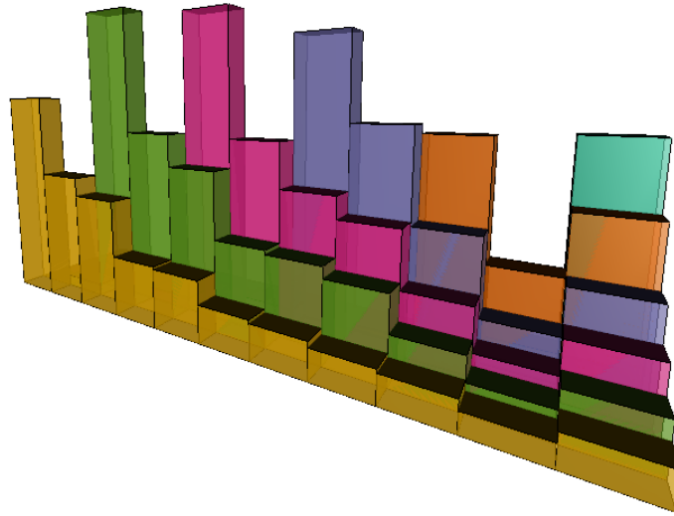


Figure 3.4: 3D version of pyramids every two years. The height of each bar depicts the number of developers that stayed for 'x' number of years and each color depicts the data for a particular year.

3D visualizations are generally deemed to be difficult to understand due to their structuring and also a user would need to view them from different orientations which adds to the complexity [24]. This could have led to difficulty in understanding the visualization.

CASE 2 -Where do Developers work ?

Location has always been of prime importance in any area of work. In the software industry, distance has never been a matter of concern because of the nature of work involved, with this being even more true for OSS as its developers

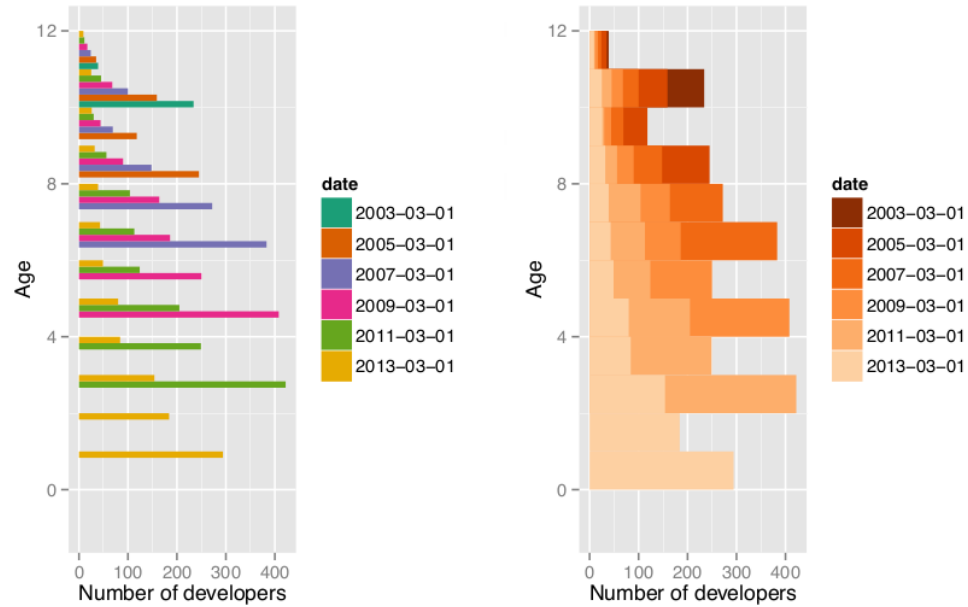


Figure 3.5: Comparison for pyramids every two years. The colors here point to data from the corresponding years and the height of the bar is the measure of the number of developers in each quarter.

are assumed to be contributing according to their convenience [22]. Bitergias main source of data for this was collected from Github and mailing lists. Because most projects hide developers full email addresses and IP addresses, certain assumptions had to be made, and geography had to be determined by time zones. They came up with two basic visualizations.

The following was the division of geography with respect to time zone -

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 basic nature of the visualizations didn't help provide a lot of detailed

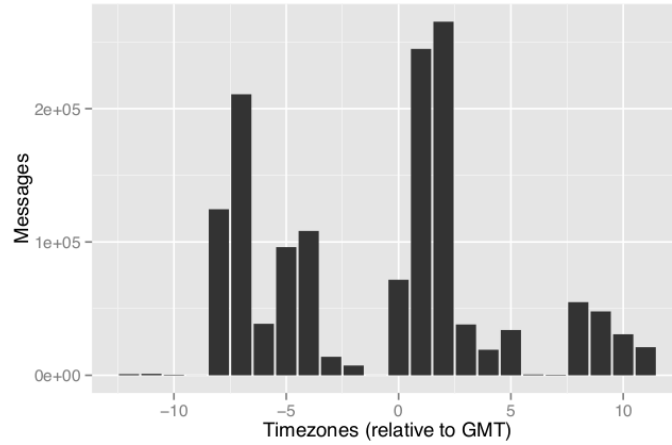


Figure 3.6: Time zone origin of messages

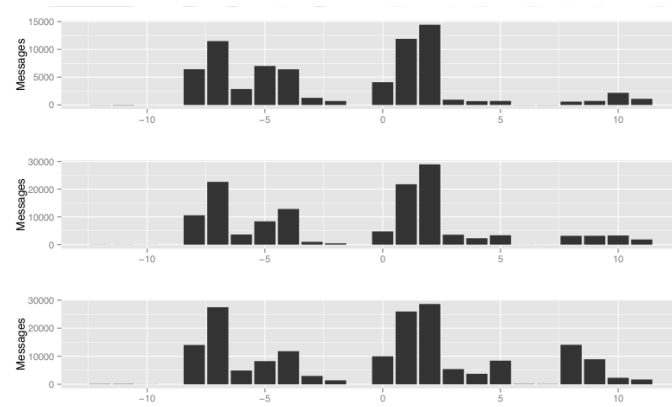


Figure 3.7: Time zone origin of messages for 2002,2007,2012

understanding about the developer distribution for an organization. They were more oriented to giving an overview of messages per time zone. The relation between commits and authors and location was not clear and needed more than a birds' eye view type of a visualization.

Chapter 4: Methodology

To help explore the previously explained cases, we looked into improvising the provided visualizations to help people understand the data in a better way. To carry out this process, datasets for 5 open source projects were collected. Different prototypes were implemented for each case and improved through informal formative evaluations with the research team. Initial visualizations were built and an iterative process of development was followed.

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 [38].
2. OpenStack : It is one of the most active open source products currently available for all types of cloud environments. It controls a large number of resources for computation, storage ,networking for a datacenter [39].
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 [40].

4. RedHat RDO: This is a community of people who are interested to use and deploy OpenStack on the RedHat Linux, Fedora distributions. It is a sub community of people who use RedHat which is one of the largest contributors to Linux [41].
5. Citrix Apache Cloudstack : This is an OSS which deploys and manages networks of virtual machines as a IaaS(Infrastructure as a Service) platform. It supports VMware, KVM, XenServer, Hyper-V [42].

4.2 Why the 5 projects?

The choice of projects were driven by a desire to cover a variety of projects. In the selected 5 projects, the Eclipse Foundation and RedHat have been around and successful for a while now. These projects also have a large developer community. OpenStack, Puppet Labs and Apache Cloudstack are relatively newer projects gaining popularity among the community. OpenStack has made its presence felt as cloud based software gathering a lot of buzz. RedHat RDO was set up as a result of the gaining popularity of OpenStack, to help people using OpenStack on Linux. This gives us a set of projects with growing communities.

4.3 Dataset

Using the tools built by Bitergia, data for the 5 organizations were extracted. The files were all in the json format. All the data collected was till February, 2015 and

holds data from 2010 to 2015. The time zone files have data from time zone -12 to +11. One important field added to the time zone files, was the approximate population for each of the time zones which was taken from the below graph.

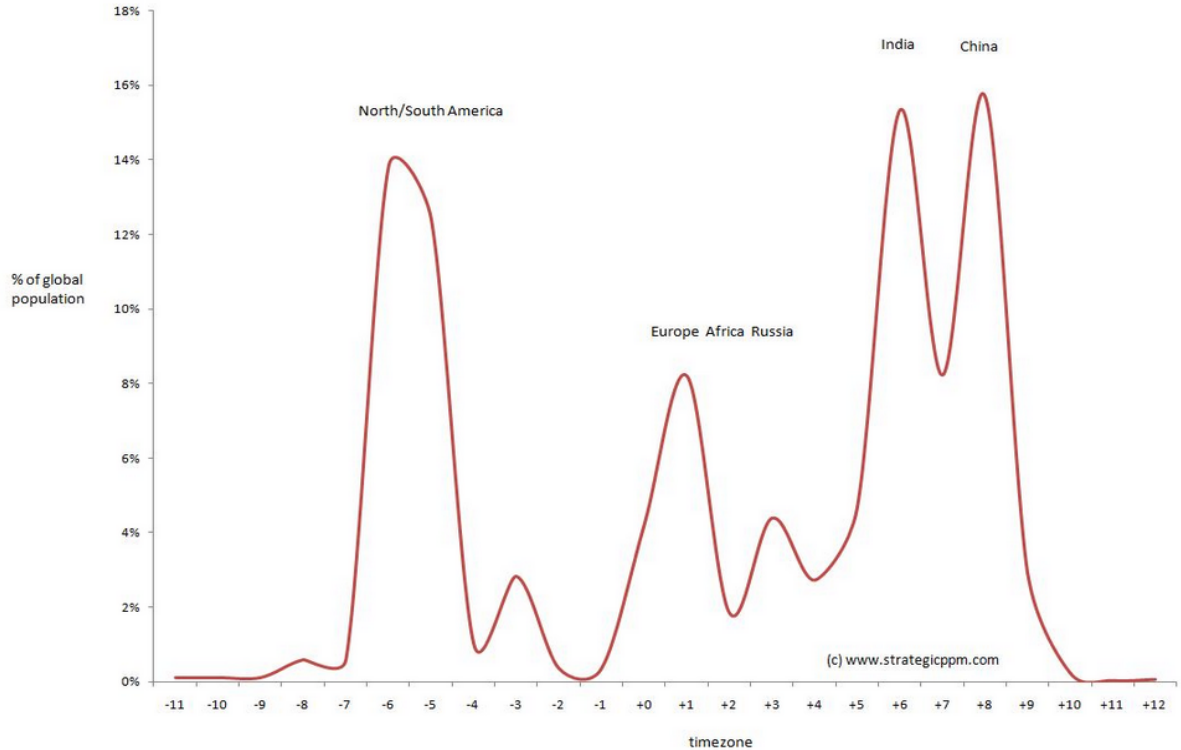


Figure 4.1: Global population by time zone

Also in order to build a time zone based visualization, a geojson file was needed which would draw the map, time zone wise and this file was made available by a researcher on Techslides [27]. As geojson files have large overhead and would take a lot of time to load, converting it to topojson format was recommended [27]. 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 Node Package Manager (NPM).

The task was then to categorize 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.

4.4 D3.js

As the visualizations needed flexibility the most ideal tool in hand was D3.js [37]. 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. Due to the large developer support available for the library, it was an ideal choice to build the visualizations. It accepts external data in the form of json, CSV or TSV. Being a javascript library, it follows the same syntax, thus making data manipulation quick. 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

The implementations done by team Bitergia are viewed to analyse the data set in hand. Figure 4.2 is used to show the experience of developers and is a bar chart. The y axis is age in quarters showing the duration that a developer

stayed/contributed to the project 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 time zones. The y axis is number of messages and the x axis is the time zones of the world. These two implementations were created out of the same datasets that were used for our implementations.

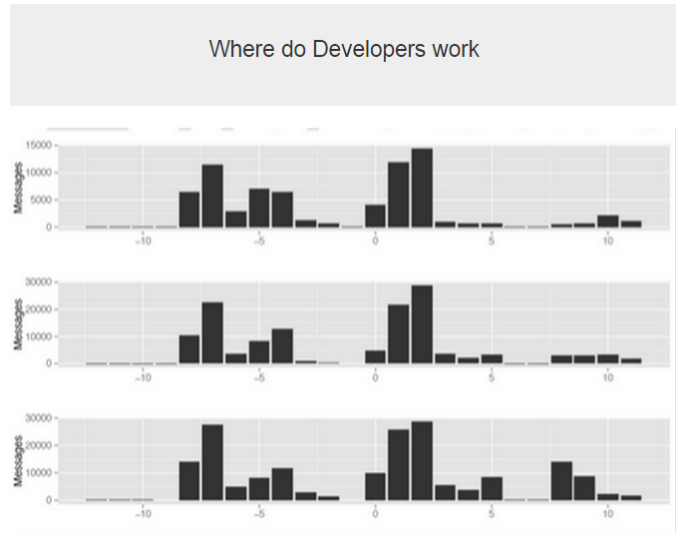


Figure 4.3: Where do Developers Work?

4.5.2 Our Implementations

Figure 4.4 is a 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.

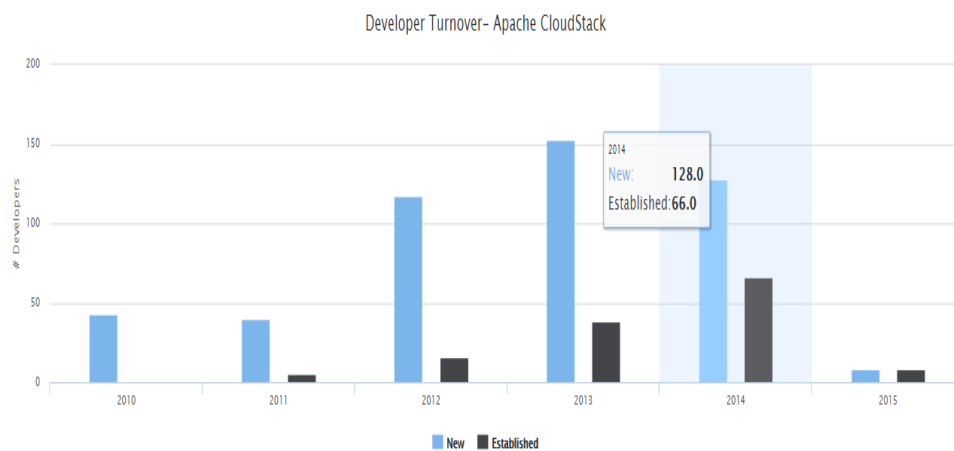


Figure 4.4: Developer Turnover

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.

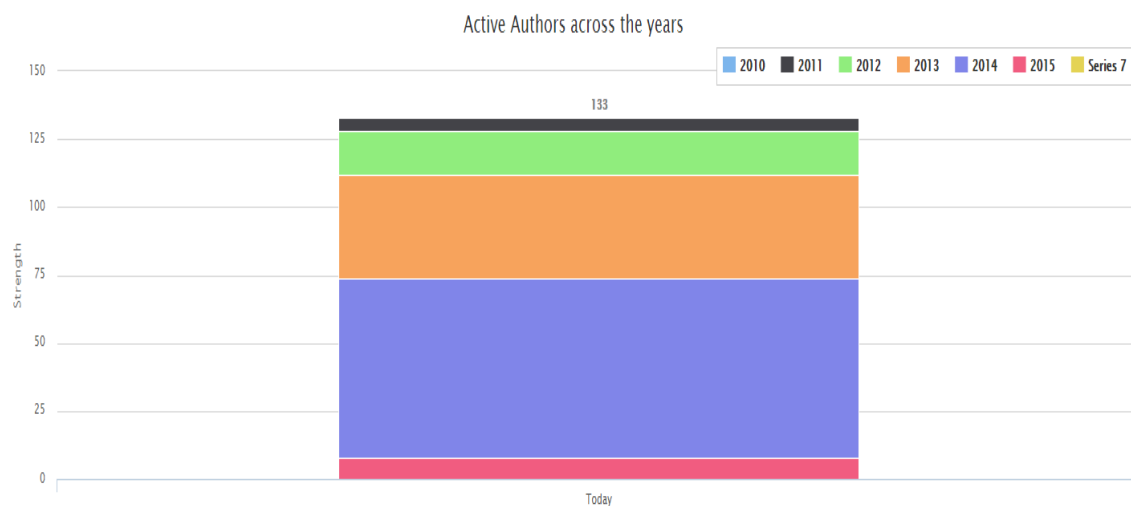


Figure 4.5: Active Authors Across the Years

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.

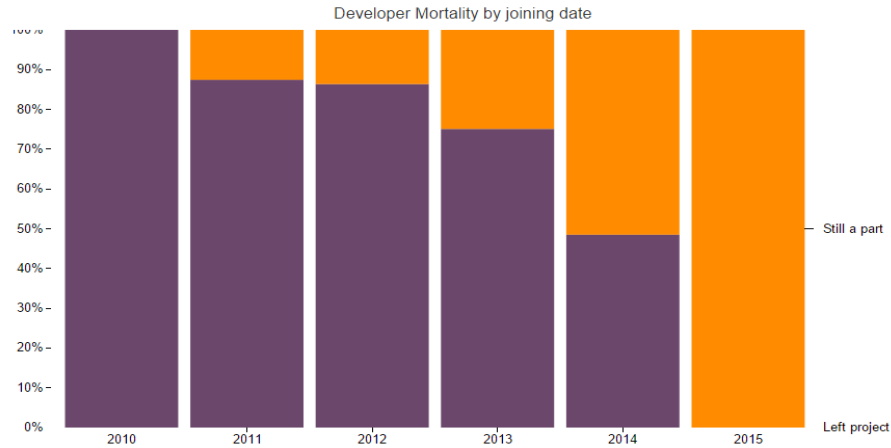


Figure 4.6: Developer Mortality by Joining Date

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 (population density). Figure 4.7 is colored according to the number of Authors present in each time zone according to the data from GitHub. Similarly Figure 4.8 is shaded with the same parameter but using data taken from the mailing lists. In each of the figures below, the orange shaded countries had no data to available. This happened with countries which had “half time zones”, like India, central Australia, Iran, Myanmar, Nepal and a small part of Canada. Also data for countries lying in time zone +12 like Russia and New Zealand was not available either.

Shaded as per Authors per TimeZone - Apache Cloudstack; Github data

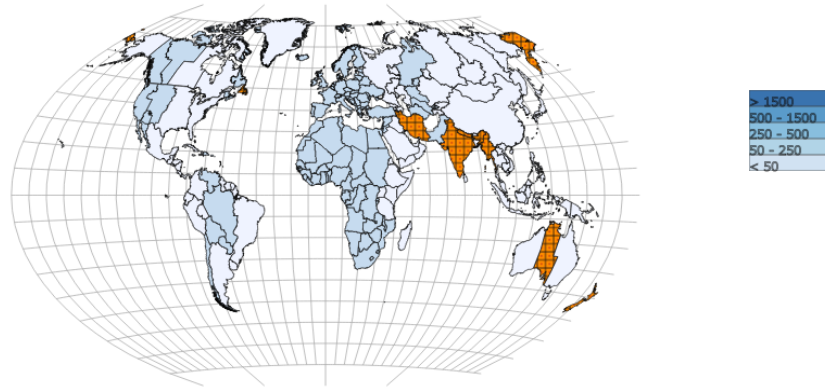


Figure 4.7: Authors per time zone - Github

Shaded as per Authors per TimeZone - Apache CloudStack; Mailing list data

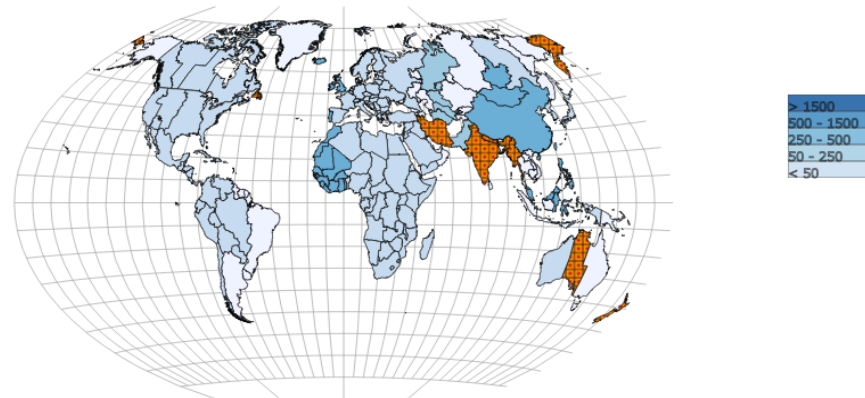


Figure 4.8: Authors per time zone - Mailing lists

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

Shaded as per Commits per TimeZone - Apache CloudStack; Github data

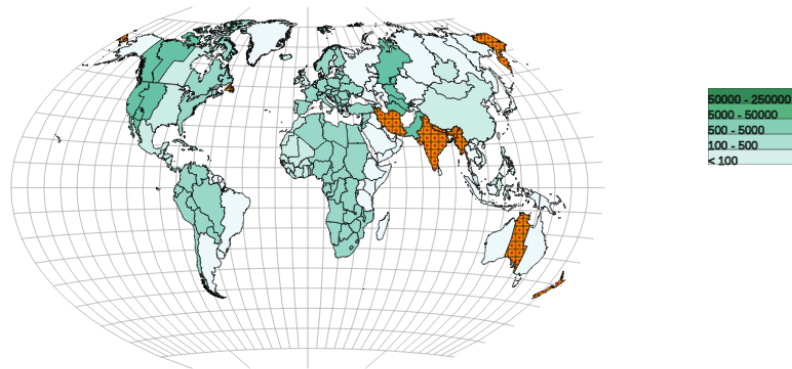


Figure 4.9: Commits per time zone - Github

Shaded as per Messages per TimeZone - Apache CloudStack; Mailing list data

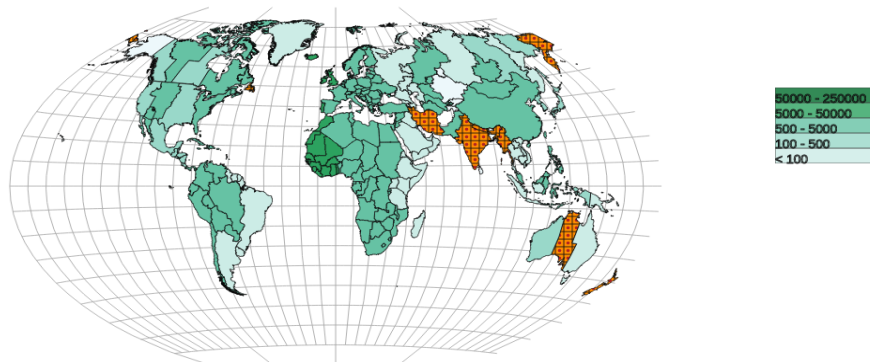


Figure 4.10: Messages per time zone - Mailing lists

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.

Contributions/Authors per TimeZone - Apache CloudStack; Github data

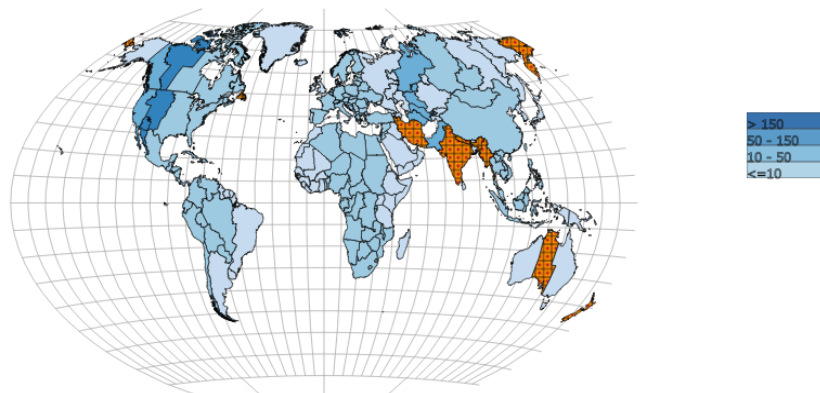


Figure 4.11: Contributions/Author per time zone - Github

Messages/Authors per TimeZone - Apache CloudStack; Mailing list data

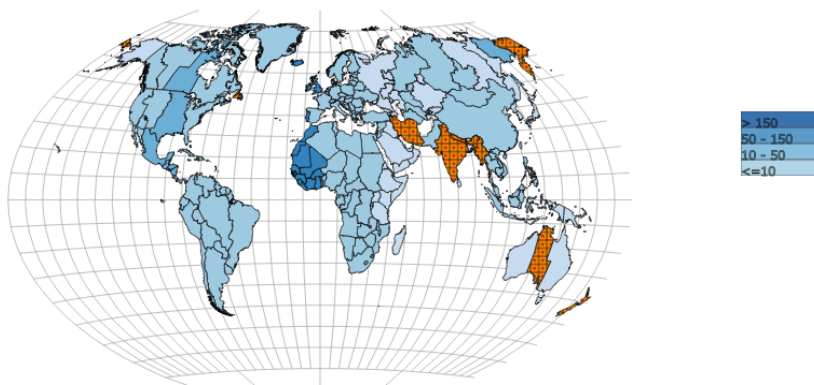


Figure 4.12: Messages/Author per time zone - Mailing lists

Commits/Population per TimeZone - Apache Cloudstack; Github data

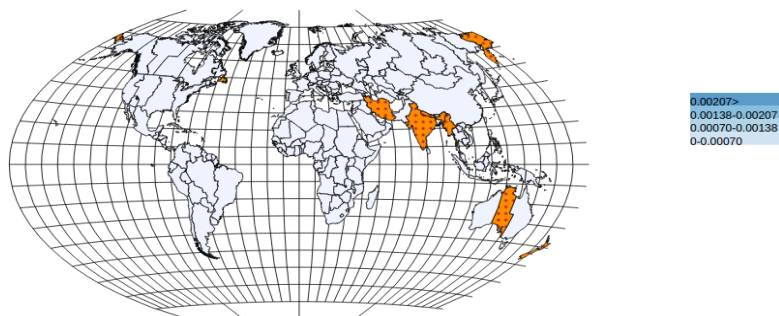


Figure 4.13: Commits/Population per time zone - Github

Chapter 5: Evaluation

In order to gauge the quality and impact of the visualizations, a focus group study was conducted. 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 [28]. Focus groups are used to gather information for discovery, benchmarking, evaluating, verifying perceptions, feelings, opinions and thoughts [29]. 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 [30].

5.1 Participants

Six participants over the age of 18 were recruited through the Oregon State University engineering mailing lists 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. Participant A3, though had a Biology background, had good software development experience as well. He had written multiple plugins for Biological data analysis in R. They were made to sign

| 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

the consent forms before the session and were asked permission to be recorded. Notes were taken and the audio part was transcribed by the researcher.

5.2 Study Session

The participants were given a nametag with their code names and scenario numbers on arrival. Three of the participants were asked to look at the visualizations done by Bitergia first, and three were given the visualizations done by us first. 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 set of visualizations. This was a within group study as all participants performed both tasks. This took up 40 minutes of the session. Once done, the participants were led to a discussion session where they were asked

some questions with regard to the visualizations. The questions that were posed 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. 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. One of the oldest research done in this area was by Tufte in 1983 [31]. 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. In 2007, Nico Marrero wrote a paper which spoke about possible metrics derived by various researchers, to evaluate the effectiveness and usability of visualizations [32]. These metrics were divided into three main categories; Mathematical Metrics, User Centric Metrics and Effectiveness Metrics. From this list, 2 metrics were picked under each category to define the usability of the visualizations.

5.3.1 Mathematical Metrics

1. Reference Context: A visualization needs to provide a context in order to be comprehensible from most viewpoints. This is the number of visible data points in any given visualization with reference to a context [33].
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 [33]. 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 be deemed complex [33].

Though these two metrics were mathematical in nature, Nico pointed that as they missed the human element of understanding, one can modify to use these metrics to certify how much the users understand the concepts and can be deemed as user centric metrics in a way as they now consider more of the whole visualization than just the quantitative aspect [32, 33]. Hence we used this aspect in our coding sessions to maintain consistency.

5.3.2 User Centric Metrics

1. Identify Clusters: The ability to group similar data points with similar qualities for better understanding [34].
2. See all features and data: The visualization should be able to show all of its underlying features and data sets without any occlusion [34].

5.3.3 Effectiveness Metrics

1. 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 [34].
2. 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 [34].

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 [35], but using pre-defined codes. Two researchers independently coded all the responses for each scenario and reached a good agreement (Cohens Kappa coefficient = 0.718 for scenario A and Cohens Kappa coefficient = 0.778 for

scenario B).

Chapter 6: Results

The questions were all open ended which gave a broad scope. Here scenario A refers to our implementations and scenario B refers to Bitergia's implementation. One thing to be noted is that this was a formative study and was performed to find if the work was heading in the right direction and not to gain consensus or state a solution to be better over the other. The idea was to gather user comments and suggestions for future work.

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 scenario B. They had done it as they could not take the help of those visualizations to answer the questions completely. This raised 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 scenario B 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 scenario A.

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 scenario B. Scenario A was easy because 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 scenario B, 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 scenario A, 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 scenario B, one participant who had knowledge of what countries fell under what time zone tried to give the solution. The other 5 said they could not answer due to lack of this knowledge. For scenario A, 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 time zone and their populations. Both of which was lacking in scenario B and required personal knowledge. They answered it better using scenario A, though not completely accurately.

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

This question needed knowledge of both countries in a time zone and their populations. Both of which was lacking in scenario B and needed personal knowledge. They answered it better using scenario A 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 scenario A 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 scenario B 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 answers for the scenario B questions as they had a tough time analysing and tying back the questions to the visualizations. As part of the coding, both researchers found that the participants felt a high reference of context with regard to the visualizations in scenario A with 5 of them making almost direct comments with this regard.

Reference Context

"Scenario A 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 scenario B 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 would not 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 time zone data as most of them did not know which country fell where. They liked how visualization in scenario A played with the colors and was intuitive which helped them as they did not 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 scenario A and once for scenario B.

See all the features and data

“Ah! Confusion no. I felt like the scenario A, the visualization was a lot more intuitive, the use of colors also helped a lot.” - A4

“I felt like scenario A 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 scenario B there was no way one can know if a certain country's data is not available and countries with half time zones are assumed to be clumped with other time zone data. But this was easily answerable in scenario A and it segregated missing data, half time zone 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 participants 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 scenario A, its very clearly said as you have that orange color with dots saying its missing and that's 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. Scenario A shows dynamics of a project over scenario B 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 scenario A visualizations. The fact that there were multiple graphs in scenario A as compared to scenario B 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 responses with each of them appearing 4 times.

Data Density

“It is always important to think about user space whenever displaying 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 scenario A 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

The table 6.1 gives an insight on how both the visualizations stood when it came to evaluating them with the metrics. Scenario A was highly appreciated for its ease of use, contextual nature and flexibility, while scenario B had one major issue of being too stagnant and low on information. The visualizations gave little but were not deemed to be ineffectual. The participants were of the opinion that if the scenario B visualizations would be combined with scenario A visualizations then

they might probably add some value to a researcher's quest.

| Metric | Scenario A | Scenario B |
|--|------------|------------|
| Reference 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

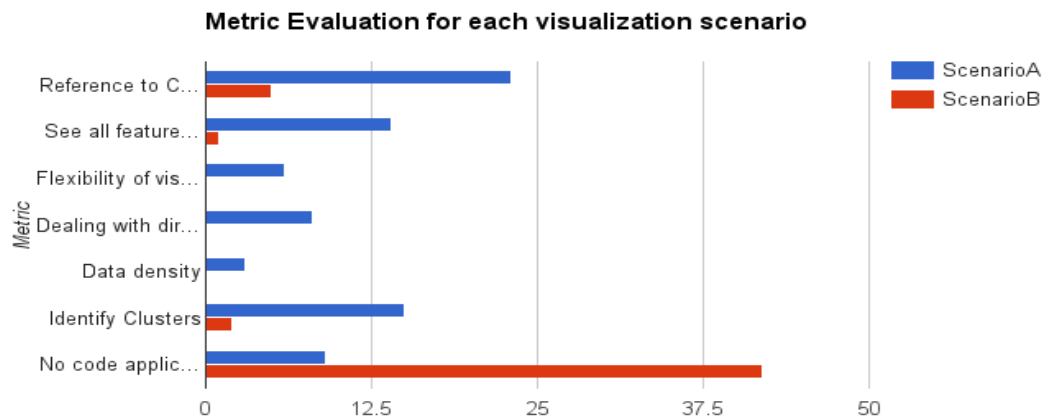


Figure 6.1: Metric Evaluation

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 scenario A and scenario B. 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. This effect was reduced 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 representative 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 misinterpretation 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 scenario A stood out plainly because it gave out the message clearly and in multiple ways. Scenario B was not thrown out by our participants, they felt it had potential too, if modified and looked at, along with the visualizations in Scenario A. Having said this there is room for improvement in the Scenario A implementations as well, as a wide variety of suggestions were obtained from the participants. Another point to be noted is, due to the small size of the group there is scope for conducting further studies to get more feedback.

Bibliography

- [1] Ghosh R A, Glott R, Krieger B and Robles G. Free/libre and open source software: Survey and study, In *Report, International Institute of Informatics, University of Maastricht, Maastricht, The Netherlands*, 2005.
- [2] The Open Source Definition - Annotated. <http://opensource.org/osd-annotated>.
- [3] Linus Torvalds quoted in The Cathedral and The Bazaar
- [4] Ten Reasons Open Source is Good for Business. <http://pcworld.com/article/209891>
- [5] David Tuma Open Source Software: Opportunities and Challenges In *CrossTalk- Journal of Defense Software Engineering*, 2005.
- [6] Sethanandha B.D , Massey B, Jones W. Managing open source contributions for software project sustainability, In *PICMET*, 2010.
- [7] 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.
- [8] Brenda Chawner Community Matters Most: Factors that Affect Participant Satisfaction with Free/Libre and Open Source Software Projects In *ACM*, 2012.
- [9] Ye Y , Kishida K. Toward an Understanding of the motivation of Open Source Developers. In *ICSE*, 2003.
- [10] Igor Steinmacher, Marco A. Graciotto Silva, Marco A Gerosa Barriers faced by newcomers to open source projects IN *OSS*, 2014.
- [11] <https://en.wikipedia.org/wiki/Anscombe>
- [12] Linux Foundation Survey <https://linuxfoundation.org/publications/linux-foundation/>.

- [13] 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.
- [14] Mockus A, Fielding R, Herbsleb J. Two case studies of open source software development: Apache and Mozilla, In *TOSEM*, 2002.
- [15] 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.
- [16] Oriol M, Oscar F D, Franch X, Marco J. Assessing Open Source Communities' Health using Service Oriented Computing Concepts In *IEEE RCIS*, 2014.
- [17] Syeed M, Hammouda I, Systa T. Evolution of Open Source Software Projects: A Systematic Literature Review, In *Journal of Software*, 2013.
- [18] 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.
- [19] 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.
- [20] 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>
- [21] Robles G , Barahona M Jesus. Geographic Location of Developers at SourceForge, *MSR*, 2006.
- [22] Takhteyev Y , Hilt A. Investigating the Geography of Open Source Software through Github, 2010.
- [23] Von Engelhardt S, Freytag A, Schulz C. On the Geographic Allocation of Open Source Software Activities. In *Jena Economic Research Papers*, 2010.
- [24] James Chiang Understanding what makes 3D Visualizations Difficult In *Rutgers Archive*, 2011.

- [25] Tuomi I. Evolution of Linux credits file: Methodological challenges and reference data for open source research, In *First Monday*, 2004.
- [26] Crowston K , Howison J. The Social Structure of Free and Open Source Software Development. In *First Monday*, 2005.
- [27] <http://techslides.com/time-zone-world-map-with-d3-and-topojson>.
- [28] Judith A. Villard. Use of Focus Groups: An Effective Tool For Involving People in Measuring Quality and Impact.
- [29] Patton M Q. Qualitative Evaluation and Research Methods.
- [30] Greenbaum T L. The handbook for focus group research.
- [31] Tufte E R. The Visual Display of Quantitative Information, In *Graphics Press, Cheshire, CT*, 1983.
- [32] Nico Marrero. Visualization Metrics: An Overview.
- [33] Richard Brath. Metrics for Effective Information Visualization, In *IEEE Symposium on Information Visualization*, 1997.
- [34] 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 Discovery*, 2001.
- [35] J. Corbin, A. Strauss. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, In *SAGE*, 2008.
- [36] Bitergia. <https://www.bitergia.com/>
- [37] D3.js. <http://d3js.org/>
- [38] Eclipse Foundation. <https://www.eclipse.org/>
- [39] Openstack Foundation. <https://www.openstack.org/foundation/>
- [40] Puppet Labs. <https://puppetlabs.com/>
- [41] Redhat RDO. <https://www.rdoproject.org/>
- [42] Apache Cloudstack. <https://cloudstack.apache.org/>
- [43] Metrics Grimoire <http://metricsgrimoire.github.io/>

APPENDICES

Appendix A: Task Responses

Scenario A 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 Cloud-stack |
| 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 - scenario A, 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 - scenario A, 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 - scenario A, 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 some 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 - scenario A, 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 time zone. |
| A2 | Time zones 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 - scenario A, case2

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

| | |
|----|--|
| A1 | No, Africa for example, has a high number commits per time zone. |
| A2 | Time zones 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 - scenario A, 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 - scenario A, case2

Scenario B 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 - scenario B, 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 - scenario B, 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 - scenario B, 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 - scenario B, 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 - scenario B, 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 - scenario B, 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 time zone. |
| A6 | Yes. |

Table A.14: Task Response 4 - scenario B, case2

