First time contributors guide for front-end frameworks

Team 7

Akshay Prasanna Kunal Satija Jose Cruz

Stevens Institute of Technology

December 12, 2021

1 Abstract

As a new developer, you may be wondering how to contribute to an open-source project, whether because you have a particular interest in open source or because it's a great way to build a portfolio for potential employers.

There has been an increase in demand for front-end developers in the last year, so in this paper, we are going to see what differentiates one project from another.

We hope that by publishing this paper, we can provide some insight into the most popular frameworks (react, vue, angular, and svelte) and their communities.

After all, one of the fundamental pillars on which open source projects can thrive is a strong community.

2 Introduction

There are multiple open-source projects out in the open, especially on Github, which house over 100 million projects [1], but we want to narrow our scope to the front-end because is an easy path to introduce a new contributor and also the impact of the collaboration will affect the lives of multiple developers and even more, the users of this applications.

In this paper we are going to focus on the main four front-end frameworks, this paper target potential first-time contributors, and can we help them to suggest which framework to choose to contribute, we are going to achieve this by taking a look into the projects and the communities that are built around them.

3 Background and Motivation

3.1 Background

When a developer wishes to start its journey as a contributor, they don't know how to begin, we suggest choosing one of the suggested front-end frameworks, it will not only increase the developer's skills but will also help with the interaction with other contributors.

This kind of interaction provides, give first-time contributors access to developers with far more experience which in turn will help them to find more opportunities outside of the scope of the project.

All the data was collected from github up to November 19, 2021

3.2 Current state of the world

We use the top four most used front-end framework: **React**, **Svelte**, **Angular** and **Vue**.

In the following table, we are going to display what is the current state of the world.

Metric	**React**	**Vue**	**Svelte**	**Angular**
Stars	176k	189k	50.7k	78k
Fork	35.4k	30.3k	2.4k	20.5k
Watch	6.7k	6.2k	874	3.1k
Contributors	401	23	445	1,498
Used By	7.6m	-	60.7	2.1m
Issues Open	885	539	618	1,847
Issues Closed	21,541	11,182	6,196	41,469
Existed Since	2013	2016	2016	2014

4 Approach

4.1 Data Fetching

For this paper we use Github API to get all the information from the projects, we get an API token, which increases the limits of their rate limit, we went through all the projects and we get all the commits and issues, and stores them in JSON files.

Commits: From commits, we get the id, the message, the author, and the date of when the commit was created.

Issues: From issues, we get the id, the message, the author, the date, the state (open or closed), and the comments, the comments are useful because this is what we are going to use for sentiment analysis.

4.2 Sentiment Analysis

We read and parse all the issues from the JSON file, each comment inside the issue has a message and an author associate with it, we split the content into words, remove all the special characters and we categorize each word into positive and negative, and with that, we assign a score to the comment.

4.3 Data Hosting

The two JSON files generated in the **Data Fetching** section are really large, the *issues* file has 405.7 MB and the *commits* file has 16.2 MB, which makes the process to query data really slow because we would need to mount all the data into memory for every request, while also need to create a script for every type of query we would like to perform to the data.

To avoid this we are going to host that information into a MongoDB instance, since the data is already in JSON we don't need to do any custom transformation, and they have a query language that supports JSON which makes it easier and faster to run queries against the data.

4.4 Edge Cases

For all the cases in which we are not able to get the data by using the MongoDB query engine, we use scripting languages to fill these gaps, this scripts will connect to the MongoDB instance, fetch the JSON and the script will do more complex queries with the data provided.

5 Experiment Design

There are two main things we want to know about the projects, to see if they are a good fit for a first-time contributor.

5.1 Is the project alive?

The first one is to figure out if a particular project is alive, by **alive** we mean a project that it's in active development and the community is active, there are examples of open source libraries that have an active community, the project still receive issues or pull request but from a roadmap perspective they already established that the project is done, so no new functionality will be added, an example of this is the library moment[2].

For this reason, we are going to see how are the commits and issues behave over time to see if there is any trend, either an increase or decline, we are going to use that as an indicator of the "liveliness" of a project.

For the data that is going to be used we are going to group them by year, and group the projects together, and see what is the trendline for each of the metrics, these are:

1. Trend of commits over the years
2. Trend of opened issues over the years
3. Trend of unique contributors per year

5.2 What is the overall sentiment of the project?

The second one is related to the community behind these projects, we want to know if they are positive, negatives, or neutral, and to which degree.

For this we are going to use the comments inside the issues, the reason we are not using commit messages is because this one are related to the code, but comments in an issue have a more social nature, so comments would be a better indicator of the project "personality" that the commit message.

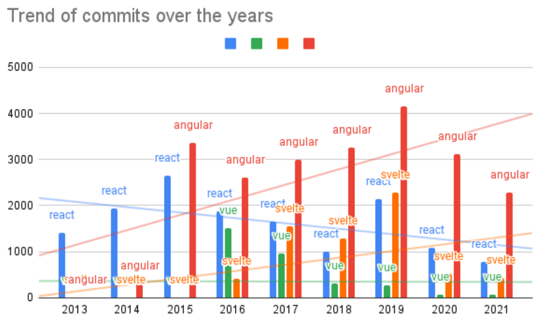
We fetch data from all the issues and each issue has a comment property that includes multiple messages sent by the contributor, collaborator, or any interested party, for each of these comments we performed a sentiment analysis on which we segmented the comment into words and group them into positive, neutral and negatives, then we see a comparative value between this and assign a score, if the word is positive we assign a positive score and if it's negative we assign a negative value, we also keep a list of all the positive and negative words for future analysis.

5.3 Answers

5.3.1 Is the project alive?

We are going to use **time** as the basis to which we are going to attach the metrics, if we see an increase over time around a project we could say that a project has an active community and the contributions, which could be issues or commits, are increasing over time, otherwise, if we see a decline we can see how steep this decline is, and raise that so the contributor takes that into consideration.

Trend of commits over the years: We use the following data set [3] that count all the commits that were performed for each project each year and we plot it in the chart below.



We can see that for React an increase in the first 3 years but after that, there has been a decline in the number of commits, on the other hand, we can see that angular started one year later after react and it surpassed it in 2015, and the rate of commits has been increasing ever since.

Vue and Svelte are the newest frameworks, both conceive in 2016, Vue's contribution has been more stable over the last few years while svelte has a trend of increasing, and the amount of commits seems to surpass Vue most years.

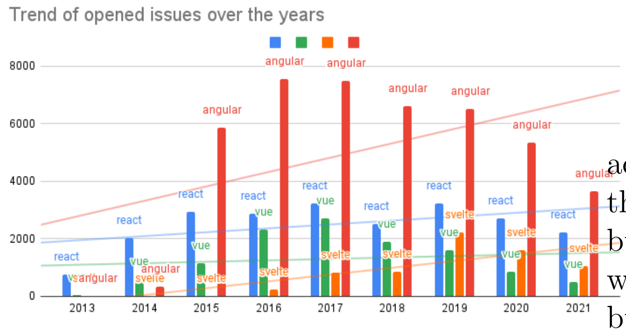
Even though react is the only project that has a declining trendline we don't think that is dying, if we take a look at the table provided in the section *Current state of the world*, react is used by 7.6 million projects.

We can't take react outside of the context in which it was born, inside Facebook, to standardize the way that developers create user interfaces, we attribute this decline to the maturity of the framework, using the Capability maturity model [4], tell us that the project is at level 5, and their focus right now is to improve performance while keeping the stability of the project.

Trend of opened issues over the years:

The last trend tells us how developers interact with the project, while this trend emphasizes how the community interacts with the project, issues are the social component in open source, at least in GitHub.

In the following chart, we did the same approach as before, we take all the data where the issues were open and we grouped them by year, and we put the projects next to each other to get any insights, below we have a chart that depicts a summary of the issues over the years, for this we use the following data set [5].

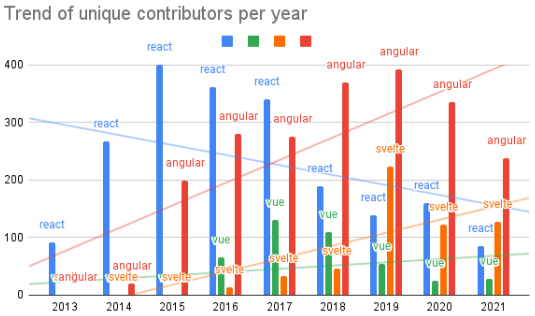


From the following chart we can see that angular has the most active community overall, from react we can see that we have an increasing trendline over the years, which contrast with the decline that it's perceived from the previous trend, which helps with the assumption that we make before that commit's alone are not a single representative metric that provides *liveliness* to a project, Vue has a slight increase over the years and has the same trend behavior that was displayed in the previous trend, svelte seems to have an increase trendline as well, surpassing Vue community in the last 3 years.

This could tell us that overall, all of these communities are fairly active which suggests that the project is alive, from a social aspect.

Trend of unique contributors per year

We already saw the number of contributions a code received, in the form of commits, and how the community interacts with each other, now are we going to see how big the community is by the number of unique developers that contribute to the project, we are going to use the following data set[6] to create the chart that is shown below.

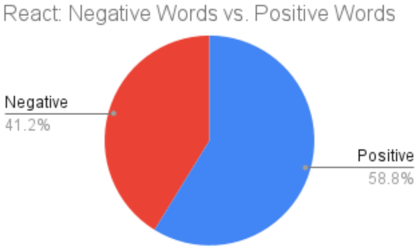


From the chart we can see a decline in React from 2015 to 2021, it could be because at the beginning more developers were joining but with the pass of time the community was getting more mature and the contributions got centralized which coincide with the trend of commits over the years, angular has received an increase in unique contributors each year which is a sign that the amount of developers is growing, svelte has seen an increase in the last years and Vue has received a fairly increase of contributors since it's inception.

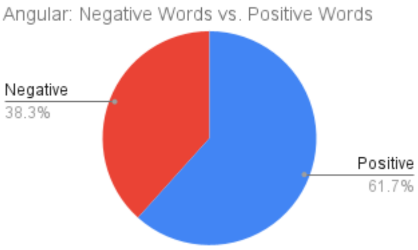
5.3.2 What is the overall sentiment of the project?

Now we are going to evaluate what are the positive and negative sentiments for each project, using the following data [7].

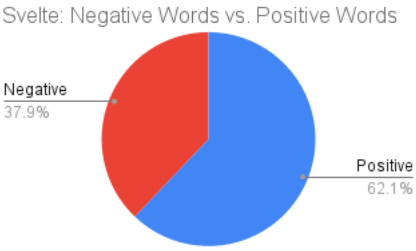
Below we are going to see a chart that depicts what is the percentage of positive and negative words used in each project.



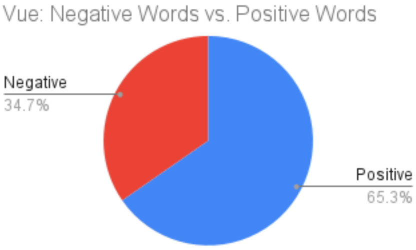
In this chart we can see that react has more positive (162,415) words that represent 58.8% compared to the negative (113,892) words, which represent 41.2%.



In the case of angular we have a more positive community than react with 280,779 positive words, which represents 61.7%, compare with the 174,500 negative words which represent 38.3%.

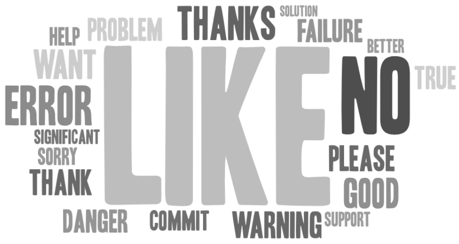


Next, we have svelte with 35,965 positive words that represent 62.1% and 21,948 negative words that represent 37.9%.

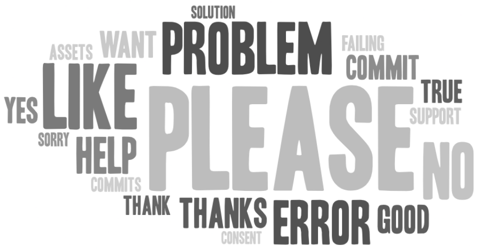


Vue has a total of 44,901 positive words, 65.3%, and 23,821 negative words, 34.7%.

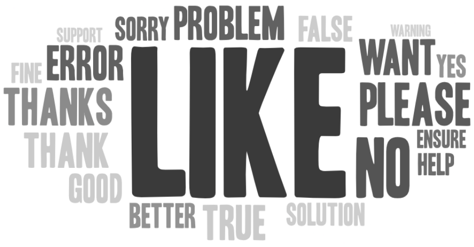
This informations suggest that all the communities are positive in general, and there are big differences between each other, now we are going to display what are the most used words, top 20, using cloud words and for each project we are going to see if it gives us an indication of what the community looks like.



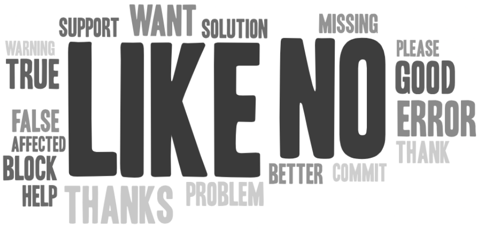
React



Angular



Vue



Svelte

From the cloud words we can see that the most use words is **Like** and **Please**, and most uses get used accross multiple projects like **thanks**, **thank**, **good**, **no**, **problem**, which suggest that there is not a specific words that difference one community from another and overall all the communities are positive, because if we take a look at the negative words are words like **no**, **danger** or **problem** which could for fixing a particular problem.

6 Analysis

Compare to svelte and Vue, **react** and **angular** has received the biggest contribution overall because they were the first one while also being backed by big companies React (Facebook) and Angular (Google), which drove early adoption.

In general, all the projects have an active community and the projects are being developed, and the community seems to use the same kind of word, so there isn't any bad choice from a community side.

If we are talking about impact, if you would like to have the highest impact we would suggest Vue because they have the lowest amount of contributors which means that it's easier to leave a mark in the project, but if you want to learn new skills and have a bigger community we would suggest Angular and React, which has the largest pool of contributors.

If the contributor would like to learn something new and a different approach than the other frameworks then we would suggest Svelte, but overall the fourth project is great for the metrics that we establish, the "liveliness" and their communities

7 Future Work

This paper was focused mainly on the top four front-end frameworks but the approach for the community could be expanded by using more social platforms that revolve around the projects, like Reddit or Stack-Overflow.

Since this one is primarily social it could give us a view of how the community interacts outside of the scope of the project, while having the usage of the project as the main topic.

The approach used during this paper could be expanded for different types of projects and libraries and try to see how their community looks like, not limiting to a first-time contributor but perhaps a seasoned contributor who would like to contribute to a project on a particular area and it has multiple projects to choose from.s

8 Conclusion

Although open source looks challenging, especially in the beginning, working on it will increase the skills of any developers while also providing a community on which the developer could rely, every open source community is different. Each community has different people in it with different roles and rewards offered.

Angular, Vue and Svelte have been increasing in their commits and community, while React has been declining but this would be a sign of maturity rather than a dying project, after all, is back by Facebook and it's an integral part of their ecosystem.

We also layout a potential use case for the project and how the community analysis could be improved to better depict how the communities interact outside of GitHub.

References

[1] Jason Warner. Thank you for 100 million repositories. https://github.blog/2018-11-08-100m-repos, accessed 2021-12-08.

[2] Moment.js. Parse, validate, manipulate, and display dates and times in JavaScript. https://momentjs.com, accessed 2021-12-08.

[3] Jose J. Cruz. Trend of commits over the years. https://github.com/jjzcru/stevens-533/blob/main/final/data/trend-of-commits-over-the-years.csv, accessed 2021-12-09.

[4] 02DCE. Software Engineering — Capability maturity model (CMM). https://www.geeksforgeeks.org/software-engineering-capability-maturity-model-cmm/, accessed 2021-12-09.

[5] Jose J. Cruz. Trend of issues over the years. https://github.com/jjzcru/stevens-533/blob/main/final/data/trend-of-open-issues-over-the-years.csv, accessed 2021-12-10.

[6] Jose J. Cruz. Trend of unique contributors per year. https://github.com/jjzcru/stevens-533/blob/main/final/data/trend-of-unique-contributors-per-year.csv, accessed 2021-12-10.

[7] Jose J. Cruz. Overall sentiment. https://github.com/jjzcru/stevens-533/blob/main/final/data/overall-sentiment.csv, accessed 2021-12-10.