

A REVIEW OF THE IMPACT OF GOOGLE CODE-IN ON OSGeo

C. Green*, V. Rautenbach

Department of Geography, Geoinformatics and Meteorology, University of Pretoria, Pretoria, South Africa –
cameron.green@tuks.co.za, victoria.rautenbach@up.ac.za

Commission V, WG V/8

KEY WORDS: Google Code-in, pre-university students, open source, OSGeo

ABSTRACT:

Many open source software communities rely on volunteer contributors and it is important to motivate, engage and retain members of the community to ensure long term sustainability of the community and software. Barrier to entry can be a problem for new developers and can stop them from contributing to large projects. It is important to mentor and guide new volunteers in an open source project and organisation such as OSGeo. The question is then raised how can open source organizations bridge this gap and bring younger developers into the organizations and ensure that they remain, in order to contribute something meaningful? OSGeo participated for a third time in Google Code-in (GCI). Google Code-in is an online competition that introduces teenagers (13-17 years) to open source development over the course of seven weeks. In the 2019 Google Code-in, there were 29 participating open source organisations, over 3000 students from more than 75 countries that completed 13 000+ tasks. During GCI, OSGeo had the lowest number of mentors in its three years of the competition but its highest number of completed tasks. Many of the submissions from the students was of a high standard and some of the task submissions got accepted into the projects. Having new developers in any open source community is key to the survival of the community and retaining them is also key to the longevity of any of the projects and it gives them time to contribute something meaningful.

1. INTRODUCTION

The majority of modern-day open source software communities rely on volunteer contributors (Steinmacher et al., 2015) and it is important to motivate, engage and retain members of the community to ensure long term sustainability of the community and software (Qureshi and Fang, 2011). The barrier to entry for new volunteers is however, often quite high and could lead them to give up (Dagenais et al., 2010). Thus, it is important to mentor and guide new volunteers in an open source project (Rautenbach et al. 2018).

Choi and Pruet (2015) conducted a study where they found the average age of open source developers to be between 30 to 49 years of age which contradicted another study conducted before 2005 by David and Shapiro (2008) where the age was between 27-30 years of age. With the age of developers being relatively high and disputed, how can open source organizations bridge this gap and bring younger developers into the organizations and ensure that they remain, in order to contribute something meaningful?

As a way of promoting open source, Google partnered with various open source organizations, such as OSGeo, Fedora Project, JBoss, and the R community, to host two annual events. Google Code-in is an online competition that introduces teenagers (i.e. between the ages of 13 to 17 years) to open source development over the course of seven weeks. In the 2019 Google Code-in, there were 29 participating open source organisations, over 3000 students from more than 75 countries that completed 13 000+ tasks (Google Open Source Blog, 2020). The second initiative is Google Summer of Code which is an initiative for university students. Google Summer of Code runs over a period of 3 months during which a student implements a new feature or

improves upon existing code for an open source project. Each student is mentored by community members. The ultimate goal of Google Summer of Code is that the students become part of the community afterwards.

The Open Source Geospatial Foundation (OSGeo) is a not-for-profit organization established in 2006 with the mission to promote global adoption of open geospatial technology (OSGeo, 2020). OSGeo has various educational initiative through GeoforAll and United Nations OpenGIS, but Google Code-in and Google Summer of Code remains the most successful.

In this paper, we will present the impact of the 2019 Google Code-in on the OSGeo community. We will provide an overview of OSGeo and Google Code-in, as well as the challenges OSGeo faced as an organization, how we overcame them, and the lessons learnt.

2. GOOGLE CODE-IN AND OSGeo

2.1. Overview of Google Code-in

Google Code-in (GCI, <https://codein.withgoogle.com>) is an annual global, online competition that introduces pre-university students between the ages of 13 to 17 years old to open source development over the course of seven weeks. Students get to work on real software packages and depending on their involvement during the competition, they can win a range of prizes from t-shirts to a trip to Google Headquarters. Since the competition first started in 2010, over 8100 students from 107 countries have participated in this competition (Google Open Source Blog, 2020).

* Corresponding author

Open source organizations can apply to participate in Google Code-in, and once selected, the organization's mentors need to produce 75+ tasks of approximately 3-5 hours ranging in complexity. These tasks can range from coding, documentation and training, outreach and research, quality assurance or design. Each task has a task description, mentor(s) that are responsible for the task, the type of task, any links to relevant information, the maximum time allowed to complete the task (i.e. between 3 to 7 days), and the number of instances available. The number of instances available refer to the number of times a particular task can be completed by the participants. Due to the nature of certain tasks, they can either have only one instance (e.g. a bug fix as once the bug is fixed, another student can no longer fix the same bug) or there can have multiple instances (e.g. a design task where the participant is required to design a certificate template) (Rautenbach *et al.*, 2018). Participants can only claim one task at a time, work on it and submit it for review.

Once a student submits a task, the mentor(s) for the task have 36 hours to review the work, but 12 hours is highly encouraged as this is a competition with prizes for students with a certain number of completed tasks. Mentor(s) can either request *more work* from the participant or *approve the task*. By requesting more work, the mentor(s) need to provide comments on how the participant can improve their submission. If a mentor feels the participant is incapable of completing the task, they can *unassign* the participant from the task and students can also *abandon* a task. Mentors also have the option to extend the deadline of the task should they deem it necessary to do so. Only once the mentor(s) have approved the task, the participant is able to claim another task and so the cycle repeats.

2.2. OSGeo's involvement in Google Code-in

The vision of OSGeo is to empower everyone with open source geospatial software. The foundation does provide financial, organizational and legal assistance to the open source geospatial community. OSGeo also serves as an outreach and advocacy organization for the open source geospatial community and a shared platform for improving cross-project collaboration. OSGeo have a wide number of open source projects currently under its umbrella. The term 'open source' applies to software that is freely distributed as well as the source code being shared (Rautenbach *et al.*, 2018). Current OSGeo projects include, and not limited to, web mapping, spatial databases, metadata catalogues, geospatial libraries, desktop applications and content management systems.

The 2019 edition of Google Code-in was its 10th anniversary, but OSGeo has only participated in the last three years (i.e. 2017 to present). In 2019, OSGeo had 16 mentors and published 110 tasks. This is a slight decline in participation numbers from the first year in 2017 where there were 20 members and 106 tasks for OSGeo and GeoForAll.

Google Code-in generally runs for 7 weeks over December and January. The dates for the 2018 Google Code-in were changed as an experiment to October – November see if the earlier slot would encourage more participation from the Southern Hemisphere. However, for OSGeo there was a decline of participation, with only 401 participants compared in 2019 to 530 participants in 2017.

In 2019, OSGeo had 16 mentors (including 4 administrators) that produced 110 tasks for 11 OSGeo projects. In 2017, OSGeo had

20 mentors, 646 tasks completed from the 106 tasks available and the tasks were completed by 530 students. 2018 saw OSGeo have 18 mentors, 471 tasks were completed from the 76 tasks available and the tasks were completed by 218 students. From this it can be seen that there is a steady decline in participation numbers, but it does pick up in 2019. In 2019 OSGeo had 16 mentors, 976 tasks completed from the 110 tasks available and the tasks were completed by 402 students. It is a debated topic but there was a decline in participation numbers for Google Code-in for 2018 across the board because of the change of dates for the competition. For 2019, the competition moved back to the same dates as 2017 and OSGeo had its greatest number of tasks completed yet.

3. AN OVERVIEW OF OSSEO'S PARTICIPATION IN 2019 GOOGLE CODE-IN

3.1. Mentors and tasks interaction

Community members were recruited to become OSGeo mentors mainly through email sent out to the OSGeo mailing lists¹. The administrators responsible for Google Code-in within OSGeo reviewed interested community members to ensure that they were an active member, as we received numerous emails from individuals that are not within the community or had any experience with OSGeo projects.

Once the mentor team was finalized, the mentors started to create possible tasks. The mentor team that consisted of 4 administrators and 12 mentors created 119 tasks in total. These tasks were not released at once but just over 75 tasks were published at the start of the competition, and the remaining tasks at various times throughout the competition. On average, each mentor was assigned to 31 tasks and interacted with 113 task instances.

Generally, two or more mentors were responsible for a task. This is important as GCI takes place over the Christmas holiday and all mentors have family responsibilities. Thus, if one mentor was not available, another mentor was able to step in and assist. Even though only known community members were selected as GCI mentors, there were 5 mentors that did not interact with any tasks or students. This can be attributed to the fact that all the mentors are volunteers and they have full-time jobs and it is not always possible to contribute.

3.2. OSGeo projects and task instances

OSGeo is an umbrella organization with numerous applications, libraries and initiatives. In 2019, the mentors created tasks for the following: GeoNode, PostGIS, pgRouting, GeoServer, GDAL, OSGeoLive, istSOS, QGIS, GRASS GIS, OpenLayers and GeoForAll. The tasks were classified and covered all the categories, see Figure 1 for a breakdown of the distribution of tasks across each category.

Google specifies the following categories for tasks:

- **Coding** tasks related to the writing or refactoring (the process of restructuring existing code) of code.
- **Documentation and training** tasks consist of creating and/or editing of documentation or tutorials to help others learn.
- **Outreach and research** tasks relate to community management, outreach, marketing or simply studying problems and recommending solutions.
- **Quality assurance** relates to the testing of code to ensure that the application or function works as expected.

¹ <https://lists.osgeo.org/mailman/listinfo>

- **Design** tasks relates to user experience research or user interface design and interaction.



Figure 1: Overview of the distribution of tasks across each category.

To gently introduce the students to OSGeo, we created 10 beginner tasks that allowed them to learn essential technologies, such as GitHub, or understand plagiarism. The GCI rules allow each student to complete 2 beginner tasks. Only when they complete the third non-beginner task, they are eligible to win a t-shirt.

The GCI platform requires mentors to specify the following information regarding each task: title, description, task duration, useful links, submission requirements, task type, tags and instances. Table 1 provides an example of the task description for a GeoNode task. There are various APIs that have been created, including an official GCI API, for the bulk importing of tasks as a CSV file.

Task Title:	GeoNode: Test the functionality of uploading of spatial layers (vector and raster) into GeoNode.
Task Description:	For this task you are going to upload spatial layers to GeoNode. Please familiarise yourself with the accepted file formats before you start. Use the following details to log into the testing instance of GeoNode: Username: gcuser19 Password: oSgEoGcI19 <i>Steps to complete this task:</i> Log into the testing GeoNode instance (https://geocatalogue.co.za), upload both a vector and raster layer (please upload GeoNode compatible documents) of your choosing. This will require two upload sessions. Set the permissions so only that you can edit the layer and verify that no errors occurred during the uploads.
Task Duration:	3 days
Useful Links:	http://geolive.co.za for more detailed instructions on how to complete this process.
Submission Requirements:	What to submit in a single PDF document: <ul style="list-style-type: none"> • Screenshots of proof that there were no errors during the uploads. • Comments of your experience, what worked for you, what didn't work for you. • Activity diagram of the steps you followed to complete this task for both uploads.
Task Type:	Quality Assurance
Task Tags:	"GeoNode", "Testing"
Task Instances:	20

Table 1: Example of a task layout for a task in GeoNode.

3.3. Tasks completed

Students are able to select any of the published tasks that they have not yet completed. Once they determine they have completed the task, they can submit their work for review. The mentors then have 36 hours to review the work and either request additional work or approve the task. The students could ask for assistance from mentors or other students through the Gitter channel that was created for the OSGeo GCI.

Over the period of 7 weeks, the 401 students completed 976 tasks. 254 of these tasks were beginner tasks. If students do not manage to complete a task, they have the option to abandon the task to either leave it completely or try again at a later stage. In total, 247 abandoned tasks were not completed.

Seven students completed more than 50 tasks with the top student completing 89 tasks, followed by 67 and 59 tasks respectively. On average, a student only completed 2 tasks, meaning only the two beginner tasks. These students will receive a certificate, but

do not qualify for a t-shirt. A more detailed breakdown of the results are available in Figure 2.

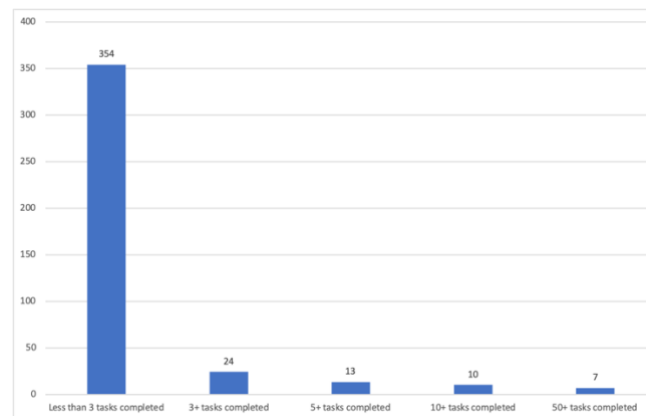


Figure 2: Number of tasks completed by students.

4. STUDENT FEEDBACK

Towards the end of the 2019 Code-in, a task was published entitled *Summarise your GCI participation experience – OSGeo*. For this task, the participants were asked to summarize their experience during the 2019 Google Code-In. They were required to write a blog post comprising of at least 300 words about: 1) the tasks they completed, 2) challenges they faced, 3) lessons they learnt, and 4) other experiences related to Google Code-in.

Most students stated their interest in geography and coding as the motivating factors for selecting OSGeo. Working with OSGeo allowed them to work with spatial data and maps while learning more about coding and open source. The general perception is that the mentors kept their best interests at heart, while motivating them throughout the process. The support students received from the mentors are essential for the success of GCI. The students consider it important that the environment is open and nurturing, rather than a dictatorship. One participant goes as far as saying “*Whenever a task was completed and approved, mentors acknowledged the participation by saying ‘Thanks for contributing to OSGeo!’ and it meant a lot. It really meant a lot to feel part of the team who is working on such a big project*”.

Most of the students indicated that the biggest thing they learnt was how to contribute to open source projects and debugging. These are important lessons for the open source community and also the tech industry in general. The participants also mentioned that they learnt not only how to contribute, but why people contribute and how software packages are built. A particular participant noticed a bug in one of the OSGeo applications and created an issue for it, followed by creating a pull request with the solution in GitHub. The pull request was accepted, and the fix is now part of the latest release.

The biggest challenge participants faced was the initial unfamiliarity with the various OSGeo applications. This is a barrier to entry, but GCI provide a gentle introduction with support from the community. Other challenges mentioned were the basic coding issues, such as debugging or software crashing.

There was no one favourite task that the participants indicated that they enjoyed, rather that they enjoyed the wide range of tasks from coding to documentation to design. They got to do tasks that they enjoyed going into Code-in as well as learning a new skill and then perfecting it which they enjoyed.

5. CHALLENGES FACED AND LESSONS LEARNED

Rautenbach *et al.* (2018) identified some lessons learnt during OSGeo’s involvement in their first Google Code-in in 2017. A few of the mentors that were mentors in 2017 were again mentors in 2019, but were any of the lessons learnt taken forward? In this section, we discuss the lessons learned and any ongoing or new challenges.

Managing the mentors’ workload

Rautenbach *et al.* (2018) stated that mentors can spend up to 30 hours a week reviewing tasks which can be overwhelming for some. They suggested that specific tasks that attract a lot of attention by the participants should be put on hold by changing the number of instances to 0 as this will allow the mentors to create a similar task with slightly different or harder requirements. This was done in the 2019 Code-in over the Christmas period to allow mentors to enjoy their Christmas break

without having the burden of reviewing task, especially if the mentor had no internet connection. Unfortunately, due to the 5 mentors that were unresponsive the workload was quite high on other mentors. In the future, mentors with previous experience will be asked to assist new mentors more closely and the administrators will need to keep a more watchful eye on the participation of mentors.

Unbalanced distribution of tasks between the categories

A concern amongst the admin team was the amount of design tasks and video documentation tasks. Design tasks are a firm favourite of the student because it is not a technical task and can be completed with relative ease in comparison. Video documentation tasks consists of making videos to explain a concept such as a popular plugin for the desktop GIS, QGIS. After reviewing all the tasks, the admin team decided that there needed to be more coding tasks. Therefore, a request was put out for more coding tasks. Once the admins were happy with the task quality and quantity, they were then uploaded to the Google Code-in Dashboard.

Clear and well documented directives and criteria for each task

In the event another admin or mentor is needed to step in to assist with the reviewing of a task, clear and well documented directives and criteria allow others to correctly review the task without the original mentor being the only one who knows what is required. This saved time and allowed for a quicker return on submitted tasks.

Detailed task descriptions

Detailed and descriptive task descriptions reduce time spent reviewing the task as the participant is more aware of what they need to submit therefore the mentor can spend less time reviewing a task. For this Code-in, mentors provided useful resources to educational material and similar tasks to guide the participants. If participants are more aware of the quality of work they need to submit, there is also less confusion and mentors can spend less time explaining the task to participants.

Following up with students when they abandon a task or are about to run out of time

Monitoring how much time a student has left on a task is important because if a mentor can see that the student is making progress but has limited time left, the mentor can extend the deadline or add more time to the task to allow the participant to finish the task and produce quality work.

Participants also abandon tasks if they feel they are not coping so mentors who see a student has abandoned a task due to not being able to complete it can offer suggestions to the participant such as extra resources to complete the task or suggesting other tasks that may prepare them for that particular task.

Preventing plagiarism

Plagiarism is an ongoing issue, with the students not being aware of what plagiarism is or why it is not good or ethical to copy from another student or sources on the internet. In the previous iterations of GCI, we asked students to including a screenshot of their terminal with their username written in it. This helped to combat plagiarism.

In 2019, we decided to be proactive and created a task to teach the participants about plagiarism and how to avoid it. Ironically, we found a couple of instances of plagiarism with this task. However, this was as the task was not well defined and considered unsuccessful. In the period until the 2020 GCI, we

will review the plagiarism task and see how we can improve the task to be more well thought out and clear.

Spam or inappropriate messages on communication platforms

In previous years, OSGeo has used an internet relay chat (IRC) as a means of supplementing the communication between the mentors and the students but this has its limitations. In the past, the IRC channel has been host to some unwelcome comments and verbal abuse from a student to other students and only the admin of the chat room had the capability to remove those who contravened the code of conduct set out by OSGeo. There was only one admin of the chat room and if they were away, no action could be taken against those who contravened the code of conduct. There were numerous instances of this occurring and it was even described as a ‘trash fire’ by one mentor. This year to manage that, OSGeo changed chat rooms and switched to Gitter. Gitter is a chat and networking platform that helps to manage, grow and connect communities through messaging, content and discovery. Gitter allows users to log in using their GitHub account which prevents spam accounts from joining the chat room as well as keeping all their personal information private as per Google’s guidelines. To prevent the issue with only one admin having admin rights of the chat room, three mentors were given admin rights to the chat room so they could ban anyone who contravened the code of conduct. The OSGeo GCI Gitter chat room is available at <https://gitter.im/OSGeo-GCI/community>.

6. CONCLUSION

In this paper, we reported on our experience participating in OSGeo’s third iteration of GCI. We reported on our experiences and also the tasks completed by the students. Since our first experience with GCI we have learned a lot, but still encountered a number of issues or challenges, such as plagiarism, unresponsive mentors, and a non-collaborative attitude of some students. Even with these challenges, we believe that GCI is worth the effort and hard work as GCI exposes students to the open source community and encourages future participation in GCI and open source. The students also get the opportunity to work with real-world scenarios and code from application in production. This is a unique opportunity for students and is invaluable for the future of an open source community, such as OSGeo. Students that participate in GCI can go further if they wish and participate in Google’s Summer of Code.

ACKNOWLEDGEMENTS

The authors would like to thank all the students, mentors, administrators and OSGeo who have participated in the 2019 Google Code-in described in the paper. We would also like to thank Google and the Google Code-in administrators for hosting and coordinating the competition.

REFERENCES

Choi, N. and Pruett, J. A. (2015) ‘The characteristics and motivations of library open source software developers: An empirical study’, *Library and Information Science Research*. Elsevier Inc., 37(2), pp. 109–117. doi: 10.1016/j.lisr.2015.02.007.

Dagenais, B., Ossher, H., Bellamy, R., Robillard, M. and De Vries, J. (2010) ‘Moving into a new software project landscape’,

Proceedings - International Conference on Software Engineering, 1, pp. 275–284. doi: 10.1145/1806799.1806842.

David, P. A. and Shapiro, J. S. (2008) ‘Community-based production of open-source software: What do we know about the developers who participate?’, *Information Economics and Policy*, 20(4), pp. 364–398. doi: 10.1016/j.infoecopol.2008.10.001.

Google Open Source Blog, 2020, The 2019 GCI Organizations!. <https://opensource.googleblog.com/2019/10/the-2019-gci-organizations.html> (1 February 2020).

Open Source Geospatial Foundation (OSGeo), 2018, OSGeo website, <https://www.osgeo.org> (1 February 2020).

Qureshi, I. and Fang, Y. (2011) ‘Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach’, *Organizational Research Methods*, 14(1), pp. 208–238. doi: 10.1177/10944281110375002.

Rautenbach, V., Di Leo, M., Andreo, V., Delucchi, L., Kudrnovsky, H., McKenna, J. and Acosta, Lara Y.S. (2018) ‘Fostering Pre-University student participation in OSGEO through the google Code-In competition’, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4W8), pp. 175–180. doi: 10.5194/isprs-archives-XLII-4-W8-175-2018.

Steinmacher, I., Graciotto Silva, M.A., Gerosa, M.A., and Redmiles, D. (2015) ‘A systematic literature review on the barriers faced by newcomers to open source software projects’, *Information and Software Technology*, 59(May 2018), pp. 67–85. doi: 10.1016/j.infsof.2014.11.001.

² <https://gitter.im>