

Study of Visual Studio Code

Practical Course — Contributing to an Open-Source Project

Michael Plainer

December 1, 2020

Visual Studio Code (VSCode) is a platform-independent code editor developed by Microsoft and released under a proprietary license. It only contains small modifications of the otherwise free and open source code base Code – OSS. This report presents the development of VSCode and Code – OSS and analyzes them in various aspects concerning FOSS development.

History

VSCode started off as a project called Monaco in 2011 with the vision of developing a code editor that could be run in the browser. For this, Microsoft decided to hire Erich Gamma as team lead, a well-known computer scientist who was also involved in the development of Eclipse [1]. Monaco was first launched to clients of the Microsoft Azure cloud enabling them to edit websites directly in the browser [2].

To make Monaco more appealing, the development team decided to switch from a web-only editor to a standalone desktop application. To achieve this, they made use of node-webkit (later called electron [3]) that allowed the team to pack and deploy JavaScript applications as native desktop apps. With this transition, Monaco became the underlying code editor that powers VSCode [1]. After this design shift, VSCode was first announced to the public in May 2015, the source code open-sourced in November 2015 and version 1.0 released in April 2016 [4].

Since then, VSCode has been continuously developed in terms of stability and functionality. As of this writing, the current version is 1.51, which naturally supports a lot more features than the initial release (e.g. WSL integration, workbenches, attached debugger, ...) listed at [5]. It is worth noting that the core team develops many of the new features in the form of plugins (e.g. the latest GitHub integration), as the goal is to keep the editor lightweight and bloat-free.

As VSCode is a dynamic open-source project, the future is uncertain. Two things the development team currently focuses on are: 1) Going back to the roots and making Visual Studio Code again available in the browser. 2) Adding support for workbenches in various programming languages that combine code and markdown text to create shareable and, most importantly, interactive projects [1] (similar to Project Jupyter [6]).

Motivation

When Erich Gamma was tasked with creating an online (code) editor, Microsoft treated this project as a greenfield project, meaning that there were nearly no restrictions and no specific direction planned. The team could focus on aspects important to them [2].

Portability and support for all major desktop operating systems has always been a driving factor during development [7]. Back when VSCode was started, developing for all desktop operating systems simultaneously was even more tedious than today, whereas websites are portable by their nature. This is why they began the development of a browser-based editor [7].

The VSCode team has always wanted to create an easily extensible and customizable experience [7]. But Erich Gamma saw an issue with existing editors and their extensibility system back then: widespread IDEs such as Eclipse can be slowed down by misbehaving plugins, making the whole development experience unsatisfactory [1]. This led to the design goal of creating a lightweight and responsive editor that only contains core features and can be easily extended by plugins [7]. Further, the communication interface between extensions and the core of VSCode has been defined in such a way that plugins should not be able to have any negative impact on the performance of the core editor while still being able to modify and tailor the experience of VSCode to user's needs [1].

One of the initial goals and also a driving factor behind VSCode was to provide Azure users with a simple online editor [2]. But the core motivation shifted to create a code editor for almost all programming languages while focusing on support for web technologies. VSCode has since then been tailored to the creation of web-based applications due to enriched built-in support for technologies such as JavaScript, TypeScript, and CSS but also for frameworks like Node.js. Still, the team tries to keep the editor general-purpose to support nearly all other environments [7].

VSCode is not only used by Microsoft internally but also by many individuals and also large, web-based companies. Facebook, one of the largest companies focusing in web-development, uses VSCode as their default development environment [8]. Considering this fact, it is safe to assume that the motivation behind VSCode to create a powerful (web) code editor has become a success.

Visual Studio Code and Visual Studio

The motivation behind VSCode may look confusing when considering that Microsoft already develops another code editor / IDE, namely Visual Studio. But VSCode was never meant to replace Visual Studio, as VSCode was designed to be a lightweight (and platform-independent) code editor for typical development tasks. It is not intended to replace any fully-featured IDE because this would be against its design goals. The fully featured Visual Studio IDE on the other hand, contains more tools and native support for more complex workflows, making it more applicable for enhanced business use-cases and programming languages that require a more sophisticated development process [9].

Governance Structure

I was neither able to find a publicly available governance structure, nor a role for every team member. An e-mail inquiry sent to Sana Ajani [10] confirmed that such a public document does not exist, but a full list of maintainers can be found at [11]. Further, I was able to find core members of the team with their corresponding roles depicted in Figure 1. Overall, the team consists of 36 members, whereas some of them also have responsibilities in other (non) open-source projects by Microsoft.

Meet the Team

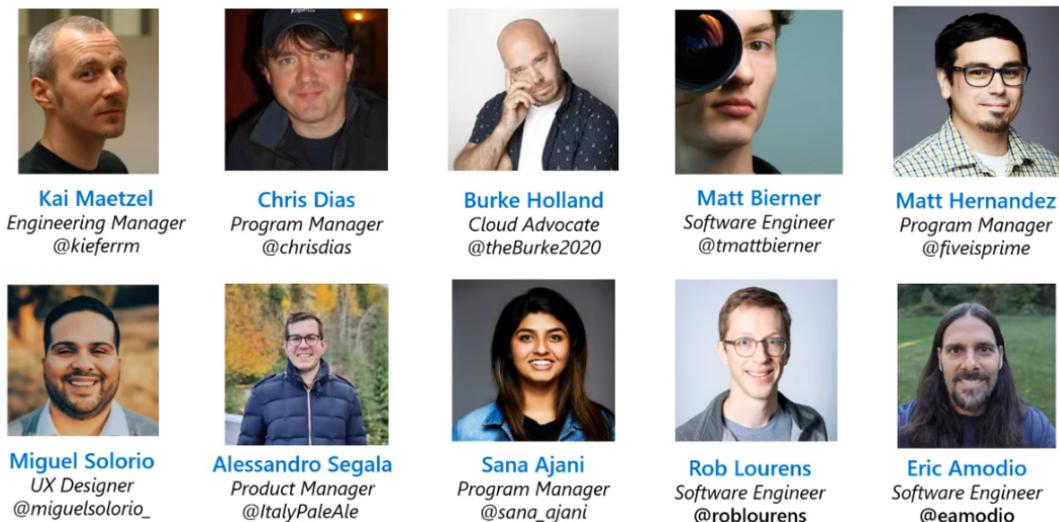


Figure 1: A slide of a Q&A talk showing essential members of the VSCode team as of May 2020 [12]. The product lead is Chris Dias, and the engineering leads are Erich Gamma (not shown in this image) and Kai Maetzel [10]. Those three can be considered the driving forces of the project responsible for guiding the team and determining future goals.

Contribution History

As VSCode is maintained by employees at Microsoft, it follows naturally that most contributions are added by those employees. As of writing, there are 1337 different contributors where about 25 people make up for nearly all changed lines (and almost account for all commits) [13].

With the help of the tool wave-gitstats [14], and some additional custom queries, I analyzed the VSCode GitHub repository to better illustrate the contribution history. This analysis and the corresponding interpretation can be seen in Figure 2 and Figure 3.

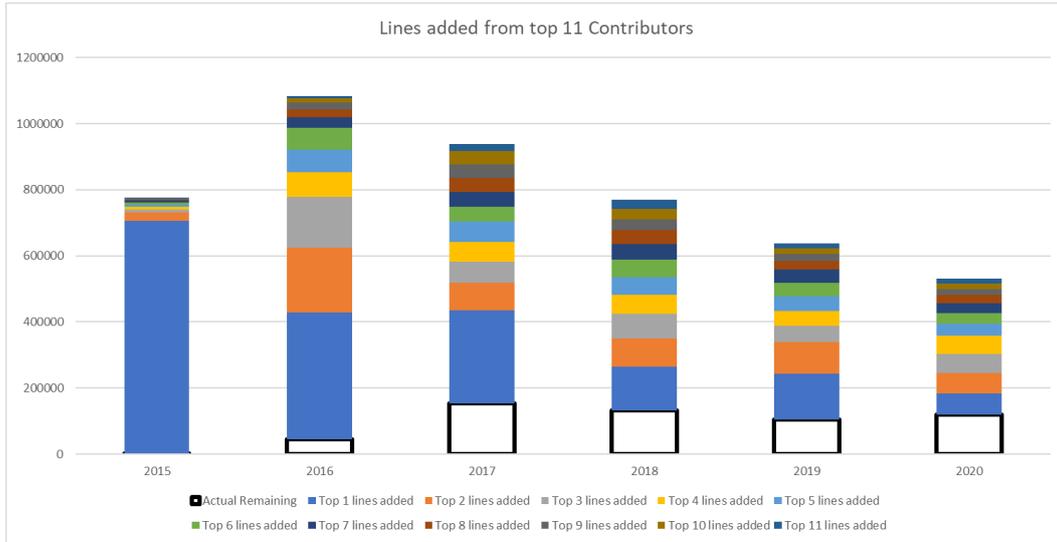


Figure 2: This bar chart shows how many people make up for the added lines in VSCode since the source code has been made available. The white box with black outline at the bottom of each year, shows the remaining lines of code that are not attributed for by the top 11 committers of this year. In the year 2015 we can see a spike of the contributor with the most lines added. This is Erich Gamma as he created the initial commit of this project. It can also be noted that over the last four years, the number of newly added / changed lines has been steadily decreasing. Also, the ratio of the most active contributors compared to all others has shrunk.

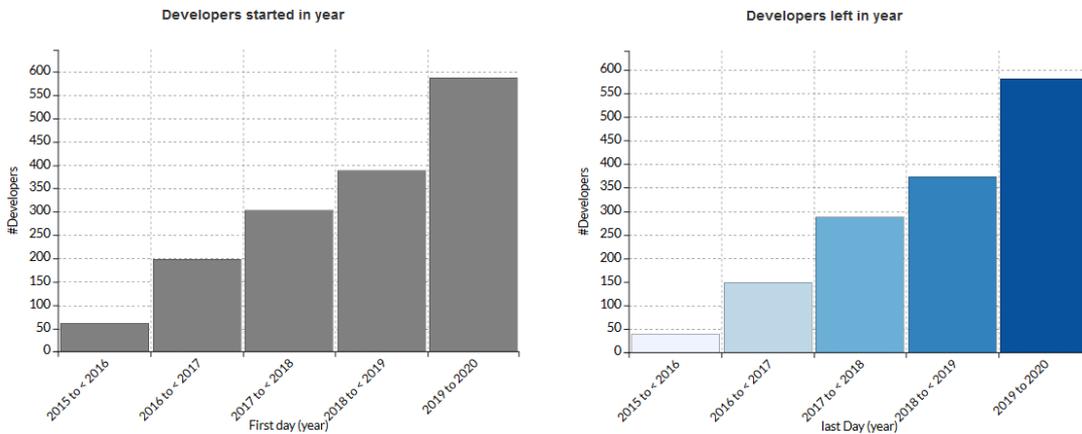


Figure 3: On the left, the developers that started in this year (i.e. first contribution), and on the right the number of developers that have left the community in this year (i.e. no contribution in the following year) are illustrated. It can be seen that the project has become more popular over the years, but most contributors only stay for a short time.

User Base

As of June 2020, VSCode has a user base of more than 11 million active users [15] and is mostly intended for and used by developers. With this information, the more interesting question arises on which type of developers use VSCode. To analyze the actual users, the StackOverflow developer survey of 2019 is presented and discussed in the next paragraphs [16]. In this survey, nearly 90,000 developers participated and were asked to fill out a questionnaire about various development related topics.

One question was regarding the used development environment. 50.7% of participants stated that they used Visual Studio Code, making it the highest ranked development environment with Visual Studio being second having 31.5%. It comes naturally that when only looking at web developers, proportionally more people use VSCode (55.6% of participants to be exact) and it was also ranked as the most used editor in this category. When looking at mobile developers and SRE (Site Reliability Engineer) / DevOps VSCode does not have as many votes but is still ranked first and second. Still, it is ranked second for Mobile Developers¹ and first for SRE / DevOps.

It should be noted that JavaScript developers (and web-related languages in general) were significantly higher represented than others. The ranking should thus be taken with a grain of salt.

Relationship to other projects

As nearly all other larger projects, VSCode has many external dependencies that it uses internally, but also some of the core functionality (e.g. language server, the file watcher, ...) is put into external GitHub repositories. All those so-called related projects are documented in [17]. Also, many additional features are delivered via plugins that reside in other repositories, some of which are not open source (and hence also not free).

Since VSCode is written in TypeScript [18], it is an interesting related project as it is also maintained and developed by Microsoft. This is especially interesting as some members of the VSCode team are also in the TypeScript team [10].

As stated in the history section, the original project started as a web editor *Monaco*. This editor is also available as its own GitHub project [19]. But this repository only contains build tools, and configurations and uses the code from the VSCode repository.

Further, it is also worth taking a look at Atom [20]. It is a text-editor similar to VSCode that is also built on web technology and used by lots of web developers. The interesting aspect is that Atom is developed by GitHub which was acquired by Microsoft. This means, that Microsoft is actively developing two very similar projects, competitors even (ignoring Visual Studio developed alongside). Both VSCode and Atom heavily rely on electron [3], a project that allows to create native desktop apps with web technologies. As most developers still prefer desktop applications, electron makes Atom and VSCode viable. Electron itself is maintained by GitHub, owned by Microsoft.

¹Although mobile developers typically use Android Studio / XCode, many apps rely on web technologies. This might be the reason why more than 53% of app developers use VSCode.

Legal, Licensing, and the Ideology of free

When discussing legal topics of VSCode it is important to distinguish between VSCode itself and the so-called *Code – OSS* project. VSCode is built on open-source rather than being open-source.

Code – OSS is released under the widely used MIT-license, making it compatible with the open source initiative [21] and is also free and compatible with the GNU GPL [22]. VSCode on the other hand is released under proprietary Microsoft Software License Terms (neither free nor open-source) available in [23].

The idea behind this separation is that Microsoft can create a custom build of the open source Code – OSS and deliver a branded product with their proprietary logo, name, telemetry, and some API endpoints while still allowing anyone to contribute, build, and distribute their custom version of Code – OSS. More reasoning behind this is described extensively by one of the developers, Chris Dias in [24].

Although this does not fully comply with the whole *free software movement*, I believe that generally this republishing is convenient for users (e.g. easy download and support) and a good way for Microsoft to build reputation, while still allowing the creation of custom builds. One major problem I see with this is that—among other things—the marketplace integration (i.e. the website to download plugins) is not publicly released and has its own proprietary license [25]. Especially since the philosophy of VSCode is to only provide base functionality and plugins are responsible to support more advanced features, this restriction drastically reduces the openness of VSCode and the freedom of users. A full list of further features that are not or only partially included in Code – OSS can be found in [25].

But since Code – OSS is free, a team of developers decided to create the project *VSCodium* that provides tools to build a FOSS version of Code – OSS. VSCodium removes any non-FOSS parts, disables telemetry, and also uses a free version for the marketplace, open-vsx [26]. Further, VSCodium builds are published regularly, making it nearly as good as the VSCode by Microsoft.

Typically, extensions that cannot be downloaded via this open-vsx integration, can be manually downloaded from the official marketplace, and installed via the `.vsix` file or have some other known workaround [27]. However, as a reddit user recently pointed out, some plugins developed by Microsoft with proprietary code check for VSCode to have a valid signature from Microsoft and prevent installation in custom builds such as VSCodium [28]. This further highlights the difference between being open-source and built on open-source. The full write up can be found in [29].

Management of Contributions

As this project is mostly maintained by employees at Microsoft, I would like to focus in this paragraph on how contributions by the community are managed. Due to its size, VSCode has elaborate guides on how contributions are handled and guidelines on how the community should interact with each other (e.g. Code Of Conduct) [30, 31, 32].

Contributions almost always begin with an issue which is either created by a maintainer or a member of the community. All issues that are created, especially those by the community, should follow the guidelines [30]. New issues are then either automatically tagged by an issue-tagger bot or by a member of the team. The person responsible for the public issue inbox and manual tagging, rotates every week [33].

If the issue is labeled as “bug” or with “help-wanted”, any member of the community is free to work on this issue, but people are advised to announce that they are working on this issue so there are no duplicate pull requests [31]. If, however, the issue has none of those two labels (e.g. features), to increase the chances of the code actually getting merged, one should ask the team if external contributions are welcome and discuss the feature. Especially architectural changes need to be discussed and evaluated beforehand [31]. And as always, the coding guidelines should be adhered when changing code [34].

Once the change has been implemented, a pull request has to be created that will be tested with an automated build pipeline and reviewed by one of the maintainers. Ideally, the change is then accepted and merged into the master with the corresponding issue being closed. Of course, there may be some further discussion needed.

First time contributors have to sign a *Contributor License Agreement* (CLA) [35] before the changes can be merged. This is an automated process and a legal protection for Microsoft that they are allowed to use and publish your code.

With every release, each contributor that helped out is specifically thanked on the release page [5].

Communication

With the maintainers being employed at Microsoft and some of them working in the same office, meetings and internal discussions occur that are generally not public. Other than that, the team has set up means of communication for the community. The following paragraph summarizes [36], where all communication channels are explained thoroughly. Links and instructions on how to join the described community are also listed there.

Users that have difficulties with the editor (i.e. something that is not a bug) or want to configure VSCode in a custom way etc. are advised to create a question on StackOverflow, as the community there is very active and eager to help. For the general community there is a Gitter chat room where they can talk about VSCode related topics. Specifically for developers (also for those developing extensions) there is a Slack workspace.

The contribution from the official maintainer team in those described channels is rather minimal, with the most participation in the Slack workspace. On the other hand, there are feature-centered discussions, where the maintainers take an integral part. Those are mostly held in the form of GitHub issues so that they are visible to the public and others can contribute.

Development Process and Release Management

The core team maintains a general plan, a roadmap, for the next 6–12 months where they describe what direction they want to take, and which features to add [37]. This plan is then implemented in monthly iterations, where each iteration is documented and regularly updated in an issue [38]. The aim for every iteration is to produce a new build that can be used by the community. This means, that roughly every month a new release is published. Each iteration can roughly be structured into three phases. 1) Monitoring the previous release, fixing crucial bugs, and planning the upcoming iteration, 2) Implementing new features according to the iteration plan, 3) Starting the “end game”.

The end game is the phase where the code of this iteration will be finalized and prepared for the next release. When all manually triggered extensive tests (called smoke tests [39]) pass, an insider build will be prepared. This build is released to the public but only those who want to be on the bleeding edge will receive those builds. This build is then thoroughly tested and monitored for any problems. Once it is deemed stable enough (i.e. 24 hours with no change whatsoever), it will be released to the public as a new full release [37].

Funding

VSCoDe is free* (as in beer²) and there is no indication of any external funding, meaning that Microsoft pays for all expenses. From a business point of view, creating an IDE / code editor in-house is not a bad idea (with the size of Microsoft at least). Employed developers can tailor it to the needs of colleagues and make it a viable tool used by many (of Microsoft’s) developers. Further, with the acquisition of GitHub, it would make sense to integrate VSCoDe / Monaco into the platform, allowing developers to quickly edit all types of programming languages in their browser. Also, in other areas, such as the Azure development platform, the integration of VSCoDe might add a lot of convenience for customers making Microsoft’s products more viable and attractive to users. Those integrations could result in indirect revenue generated by VSCoDe.

Conclusion and Personal Opinion

Visual Studio Code has become one of the most used (code) editors available. The fact that Microsoft decided to release the underlying source code is definitely a step into the right direction, making developing plugins easier and developers more trusting about the code and the telemetry sent to Microsoft. But since VSCoDe is not free, and Microsoft still applies some questionable business practices (e.g. prevent installing plugins on non-official VSCoDe instances), anyone supporting the free software initiative should take a look at a free version of VSCoDe, like VSCodium.

²Code – OSS is also free as in speech.

References

- [1] Erich Gamma, Asim Hussain, and Simona Cotin. *The History of Visual Studio Code*. Microsoft Channel 9. May 26, 2020. URL: <https://channel9.msdn.com/Events/Build/2020/BDL134> (visited on 11/19/2020).
- [2] Mary Jo Foley. *Microsoft's Browser-Based Dev Toolbox: How 'Monaco' Came to Be*. ZDNet. Nov. 18, 2013. URL: <https://www.zdnet.com/article/microsofts-browser-based-dev-toolbox-how-monaco-came-to-be/> (visited on 11/19/2020).
- [3] GitHub. *Electron*. Electron, Nov. 22, 2020. URL: <https://github.com/electron/electron> (visited on 11/22/2020).
- [4] The VS Code Team. *Visual Studio Code 1.0 Release*. Visual Studio Code. Apr. 14, 2016. URL: <https://code.visualstudio.com/blogs/2016/04/14/vscode-1.0> (visited on 11/19/2020).
- [5] Microsoft. *Updates*. VSCode. URL: <https://code.visualstudio.com/updates/> (visited on 11/24/2020).
- [6] Jupyter. *Project Jupyter*. URL: <https://www.jupyter.org> (visited on 11/19/2020).
- [7] The VS Code Team. *Why Did We Build Visual Studio Code?* GitHub. Nov. 6, 2020. URL: <https://github.com/microsoft/vscode-docs/blob/27b4fd1408af3d2736aeaf5d64fa811b0b292508/docs/editor/whyvscode.md> (visited on 11/19/2020).
- [8] Hassan Mehedi. *Facebook Adopts Visual Studio Code Internally Following Microsoft Partnership*. Thurrott.com. Nov. 19, 2019. URL: <https://www.thurrott.com/dev/222366/facebook-adopts-visual-studio-code-internally-following-microsoft-partnership> (visited on 11/19/2020).
- [9] The VS Code Team. *Visual Studio Code Frequently Asked Questions*. Nov. 6, 2020. URL: <https://code.visualstudio.com/docs/supporting/faq> (visited on 11/19/2020).
- [10] Sana Ajana. *Sana.Ajani@microsoft.Com*. sana-ajani.me. URL: <http://www.sana-ajani.me/> (visited on 11/22/2020).
- [11] Kai Matzel. *2020 Issue Grooming - Review, Categorization, Assignment*. GitHub. Oct. 10, 2020. URL: <https://github.com/microsoft/vscode/issues/108472> (visited on 12/01/2020).
- [12] Microsoft Developer, director. *Ask the Team: Visual Studio Code — COM01*. May 25, 2020. URL: <https://youtu.be/VRiaDG--A4k?t=159> (visited on 11/19/2020).
- [13] Microsoft. *Contributors*. GitHub. URL: <https://github.com/microsoft/vscode> (visited on 11/20/2020).
- [14] SuRaMoN. *Weave-Gitstats*. Aug. 11, 2020. URL: <https://github.com/SuRaMoN/weave-gitstats> (visited on 11/20/2020).
- [15] The VS Code Team. *The next Phase of the Go Experience*. Visual Studio Code. June 9, 2020. URL: <https://code.visualstudio.com/blogs/2020/06/09/go-extension> (visited on 11/22/2020).
- [16] Stack Overflow Team. *Stack Overflow Developer Survey 2019*. Stack Overflow. 2019. URL: <https://insights.stackoverflow.com/survey/2019#development-environments-and-tools> (visited on 11/22/2020).
- [17] Microsoft. *Related-Projects*. VSCode Wiki. June 7, 2020. URL: <https://github.com/microsoft/vscode/wiki/Related-Projects> (visited on 11/22/2020).
- [18] Microsoft. *TypeScript*. Microsoft, Nov. 22, 2020. URL: <https://github.com/microsoft/TypeScript> (visited on 11/22/2020).
- [19] Microsoft. *Monaco-Editor*. Microsoft, Nov. 22, 2020. URL: <https://github.com/microsoft/monaco-editor> (visited on 11/22/2020).
- [20] GitHub. *Atom*. Atom, Nov. 22, 2020. URL: <https://github.com/atom/atom> (visited on 11/22/2020).

- [21] Opensource.org. *Licenses & Standards*. Open Source Initiative. URL: <https://opensource.org/licenses/> (visited on 11/24/2020).
- [22] Free Software Foundation. *Various Licenses and Comments about Them*. GNU Operating System. May 7, 2020. URL: <https://www.gnu.org/licenses/license-list.en.html#SoftwareLicenses> (visited on 11/24/2020).
- [23] Microsoft. *License - Visual Studio Code*. VSCode. URL: <https://code.visualstudio.com/license> (visited on 11/24/2020).
- [24] Chris Dias. *Menu License Links to Non Open Source License*. GitHub. Dec. 3, 2015. URL: <https://github.com/Microsoft/vscode/issues/60#issuecomment-161792005> (visited on 11/24/2020).
- [25] Microsoft. *Differences between the Repository and Visual Studio Code*. VSCode Wiki. Mar. 21, 2020. URL: <https://github.com/microsoft/vscode/wiki/Differences-between-the-repository-and-Visual-Studio-Code> (visited on 11/24/2020).
- [26] VSCodium. *Vscodium*. VSCodium, Nov. 24, 2020. URL: <https://github.com/VSCodium/vscodium> (visited on 11/24/2020).
- [27] VSCodium. *VSCodium Docs*. GitHub. Oct. 30, 2020. URL: <https://github.com/VSCodium/vscodium/blob/16eb6e4c833b2b1cb02509f963ed18a70919c4a0/D0CS.md#proprietary-debugging-tools> (visited on 11/30/2020).
- [28] Jake Bailey. *Can't Find and Install Pylance*. GitHub. Oct. 13, 2020. URL: <https://github.com/microsoft/pylance-release/issues/483#issuecomment-707916515> (visited on 12/01/2020).
- [29] /u/TheEdgeOfRage. *VS Code Developers Prevent Running the New PyLance Extension on Open-Source Builds of VS Code*. reddit. Nov. 26, 2020. URL: https://www.reddit.com/r/linux/comments/k0s8qw/vs_code_developers_prevent_running_the_new/ (visited on 11/28/2020).
- [30] Microsoft. *CONTRIBUTING.Md*. GitHub. Sept. 15, 2020. URL: <https://github.com/microsoft/vscode/blob/c2d3d7f8574dc2bbac62f1319e7fe07513c70365/CONTRIBUTING.md> (visited on 11/24/2020).
- [31] Microsoft. *How to Contribute*. VSCode Wiki. Nov. 17, 2020. URL: <https://github.com/microsoft/vscode/wiki/How-to-Contribute> (visited on 11/24/2020).
- [32] Microsoft. *Code of Conduct*. Microsoft Open Source. URL: <https://opensource.microsoft.com/codeofconduct/> (visited on 12/13/2020).
- [33] Microsoft. *Issue Tracking*. VSCode Wiki. Aug. 4, 2020. URL: <https://github.com/microsoft/vscode/wiki/Issue-Tracking> (visited on 11/24/2020).
- [34] Microsoft. *Coding Guidelines*. VSCode Wiki. July 29, 2020. URL: <https://github.com/microsoft/vscode/wiki/Coding-Guidelines> (visited on 11/24/2020).
- [35] Microsoft. *Contributor License Agreement*. VSCode Wiki. Feb. 12, 2020. URL: <https://github.com/microsoft/vscode/wiki/Contributor-License-Agreement> (visited on 11/24/2020).
- [36] Microsoft. *Feedback Channels*. VSCode Wiki. Oct. 23, 2020. URL: <https://github.com/microsoft/vscode/wiki/Feedback-Channels> (visited on 11/25/2020).
- [37] Microsoft. *Development Process*. VSCode Wiki. Aug. 26, 2020. URL: <https://github.com/microsoft/vscode/wiki/Development-Process> (visited on 11/25/2020).
- [38] Microsoft. *Iteration Plans*. VSCode Wiki. Nov. 16, 2020. URL: <https://github.com/microsoft/vscode/wiki/Iteration-Plans> (visited on 11/25/2020).
- [39] Microsoft. *Smoke Test*. VSCode Wiki. Feb. 28, 2020. URL: <https://github.com/microsoft/vscode/wiki/Smoke-Test> (visited on 11/25/2020).