01.12.2020

Lab Course Contributing to an Open Source Project

Preliminary Report Visual Studio Code Leonard Goldstein

Introduction

This preliminary report deals with the Open Source project `microsoft/vscode`, also known as Visual Studio Code, which is a code editor written mainly in TypeScript. Created in the context of the Lab Course: Contributing to an Open Source Project at TUM, in this report I intend to inform and motivate my contributions and possibly the contributions of others to that project. Visual Studio Code, which is used by software developers for development in many different languages, but especially for web development, is one of the most popular Open Source projects in existence. Nevertheless, only ten years earlier, few people would have believed that the release of this kind of product would be possible for this kind of company. Therefore, the report highlights the circumstances that lead to the project. Also, because Visual Studio Code is a project consisting of many parts, the report explains what project parts can be found where and how the contribution process works.

Project History & Motivation

Microsoft's relation to software development and Open Source

Microsoft has a long history of creating products targeted at developers. The first product under the company name Micro-Soft was Altair BASIC (1975) (Gates), a BASIC interpreter for the Altair 8800 computer. Since then, developers were some of the most important customers of the company, which might be what Microsoft's then-CEO Steve Ballmer probably tried to communicate when he famously chanted "Developers, Developers, Developers, Developers" energetically to his employees in an internal event.

Until about the middle of the 2010s, Microsoft's developer-centered software was proprietary and often commercial. One of Microsoft's most notable series of developer products is Visual Studio, an integrated development environment (IDE) for various programming languages. Depending on the edition, retail prices for the first version, Visual Studio 97, were between 999$ and 1499$ (Thurrot, 1997). Prices for newer versions varied but eventually went further down: Since 2005, Microsoft even releases free-of-charge editions ("Express"), which contain, since 2014 ("Community") most features of the commercial variants.

There were many reasons for that price/value shift. One might have been the availability of more good FLOSS software in the same sector – a point that caused Microsoft headaches:

Ballmer reportedly saw himself as an enemy of FLOSS and criticized the GNU GPL and especially Linux as “cancer that attaches itself in an intellectual property sense to everything it touches” (Greene, 2001). Another reason, especially for the later versions, was a huge shift in Microsoft’s focus. Around the time when Satya Nadella became CEO in 2014, Microsoft started expanding in fields like cloud business, in the consumer market for example with Office 365 and OneDrive, and in the developer market with Azure. The role of development software subsequently changed: Instead of being the product, it became more of a gateway to the products.

That opened new possibilities for licensing and community involvement: During this shift, Microsoft started actively engaging with many Open Source communities and changing the licensing of a lot of development software. Important examples include the founding of the .NET Foundation and the subsequent Open Source release of .NET (Paoli, 2014), the acquisition of GitHub Inc. (Microsoft acquires GitHub, 2018), and joining the Open Source Initiative. At the time of writing, Microsoft has over 3800 repositories on GitHub under the Microsoft page alone and is the single biggest contributor to Open Source projects in the world (Warren, 2020). Due to Microsoft’s size and history with open source, not all developers and activists were without concern about that change – when the GitHub Acquisition was initially announced, some developers left the platform, out of concern that the hosted software’s freedom might be in danger. That backlash was, however, temporary.

From Monaco to Code

When expanding in the cloud market, it was a logical step for Microsoft to work on the ease of access to online services through web interfaces. Especially in products targeting developers, in many cases, a way to show code to developers and let them edit it in the browser. Even though there were already working Open Source browser-based code editors, like CodeMirror since 2007, Microsoft asked Erich Gamma, who is best known as a co-author of „Design Patterns: Elements of Reusable Object-Oriented Software“, to lead a project with the intent to create another alternative (Dias, Juarez, & Gamma, Meet the Visual Studio Code team, 2015). Reasons for the creation of an own alternative are not known publicly, but various points might have contributed to that: When Gamma was hired in 2011, there was not yet a performant and generally established project. Projects with support for fewer languages existed. And bigger projects like CodeMirror just started to get more performant, CodeMirror released their completely rewritten 2.0 version in that same year (Haverbeke, 2020). Microsoft has established a culture of “eating their own dog food”, that is, using the products they develop in their own workflows. Vice versa, it is also not atypical for Microsoft - to stay in the same metaphor - to produce a lot of dog food to eat. From Microsoft’s pre-cloud-focused perspective, it made more sense to be the owner of everything they use.

This project, codenamed Monaco, was a success story for the team right away: As it turned out, many different Microsoft products could profit from an HTML/JavaScript-based Code Editor. Monaco was adapted in the Internet Explorer F12 tools, Visual Studio Online (now

Azure DevOps), and for Code previews in SkyDrive (now OneDrive) (Dias, Juarez, & Gamma, Meet the Visual Studio Code team, 2015).

Visual Studio (IDE) is a huge piece of software: Visual Studio 2019, for example, requires up to 210GB of space (Visual Studio 2019 Product Family System Requirements, 2019). In some scenarios, this might be reasonable, but many setups, especially when deployed to a remote server, don't require many tools apart from what Gamma calls the "inner loop": Syntax highlighting, version control, and debugging tools. Other tools would, in many scenarios, only slow down the user. That made the extension of Monaco to a complete code editor, called Visual Studio Code, a natural consequence.

The way to Open Source

Visual Studio Code is one of three similar projects released around 2015. Adobe Brackets, the first stable version released in November 2014, and GitHub's Atom, stable since June 2015. The success of Atom and to some extent Brackets was determined by various selling points. Both editors were, in comparison to the previous generation of editors, highly customizable and extensible: The editors were written with web technologies, and web developers were one of the main target groups of lightweight code editors. Also, the editors were Open Source, a fact that most certainly attracted many developers.

While the decision to release Visual Studio Code as Open Source Software in November 2015 probably surprised many people, it was a logical consequence. Opening the Source was perhaps necessary to not fall behind the – otherwise similar – Atom and Brackets editors. Also, Gamma's team already were Open Source contributors, for projects that Visual Studio Code builds upon, such as Electron, and for language support OmniSharp and Roselyn (Dias, Juarez, & Gamma, Meet the Visual Studio Code team, 2015). Given the fact that Visual Studio Code was already intended as a free-of-charge application, there were no severe reasons not to release the code as Open Source.

From that point in time, Visual Studio Code was a huge success: In the Stack Overflow Developer Surveys, Visual Studio Code wasn't listed yet in 2015, but already was used by 7.5% of participants in 2016 and surpassed the slightly less performant Atom with 24% in 2017. In 2019, over 50% of all participants used Visual Studio Code (Developer Survey Results 2019, 2019). Visual Studio Code has over 19100 individual contributors now (containing over 1300 code contributors), more than any other project on GitHub (The State of the Octoverse, 2020).

In 2018, Microsoft acquired GitHub. Microsoft announced that they were going to continue working on GitHub's Atom editor, independently from Visual Studio Code's development (Foley, 2018).

Licensing

When downloading Visual Studio Code product as a binary release, it is licensed under the proprietary Microsoft Software License (MICROSOFT SOFTWARE LICENSE TERMS: MICROSOFT VISUAL STUDIO CODE, 2020). The code is, however - without some Microsoft specific customizations - licensed under the MIT license, and available on GitHub at https://github.com/microsoft/vscode/. Although this dual way of distribution is often criticized in the Free/Open Source world, it is not uncommon for projects of big companies (compare for example Google Chrome/Chromium). For better differentiation, the main repository is also referenced as "Code – OSS". To obtain an Open Source version of Visual Studio Code, one can compile the project on his own, or obtain an MIT-licensed binary via a third-party project, such as `vscodium`. Also, Open Source builds are offered in many Linux package managers, such as with pacman (community) as `code` for Arch and similar Linux distributions.

Funding

Visual Studio Code is not a project that intends to generate profit directly. Microsoft indirectly profits from the usage of Visual Studio Code, because it makes other paid services by Microsoft, such as GitHub for enterprises and many products from the Azure product range accessible easier. Therefore, Microsoft finances the necessary servers and pays the main contributors, who are employees at Microsoft.

Structure and Project Relations

Visual Studio Code consists of more parts than just the `microsoft/vscode` repository: Many components of the project are developed separately in separate repositories, most notably the standalone editor component Monaco, various debuggers, linters, the standard themes, and even documentation.

It builds upon other Open Source projects, such as the Electron framework, which makes it possible to develop desktop applications with web technologies. Electron includes the Chromium rendering engine and the Node.js runtime. It has over 1000 dependencies according to the `yarn.lock` file, which sounds much, but isn't an uncommon number for big TypeScript projects

Visual Studio Code and its related projects are not just Open Source consumers, but also suppliers, as other projects use parts of the project. Microsoft products such as Azure DevOps and OneDrive used Monaco. Some projects use big parts of Code – OSS, such as `cdr/code-server`, which is a version of Visual Studio Code that can be accessed in a browser.

Microsoft is working on an (at this point proprietary) solution for Visual Studio Code in the browser too, with GitHub Codespaces. The service currently in early access allows developers to work on and run GitHub projects in the browser, without having to download or set up the project.

What distinguishes Visual Studio Code from many other similar programs is its broad range of extensions. Over 22000 extensions are available for Visual Studio Code in its marketplace, many of them are Open Source too. Popular extensions include Microsoft's Live Share extension pack enables users to collaborate on code in real-time and Eric

Amodio's GitLens, which most importantly displays information about recent commits right in the code editor component.

Overview over the `vscode` repository

Maintainers and the Community

The core team behind Visual Studio Code works, led by Erich Gamma, in Zurich, although some members work from Microsoft's headquarters in Redmond, Washington, or remote too. Today, the core team consists of about 36 people (Maetzel, 2020). Most visible in media surrounding Visual Studio Code probably is the project's principal program manager, Chris Diaz, who works from Redmond, and so do many other members in management positions, such as a second technical lead, Kai Maetzel. Most of the more technical conversation with the community, in GitHub Issues, happens with Microsoft's software engineers directly, which, in slightly more cases, work from Switzerland: 7 of the top 10 contributors, measured in commits, are working from Zurich (Contributors, 2020).

Together with the employees working on the language extensions that are bundled in a default installation as well, there are about 100-150 people working on Visual Studio Code full-time, even more, if one counts in people working on other extensions, such as those for GitHub (Dias & Ajani, Ask the Team: Visual Studio Code I COM01, 2020).

While the 27 persons with the most commits all work or have worked for Microsoft, further down the list there are many external contributors, some of which have contributed over 100 commits and/or over 1000 lines of code. Many external contributors contribute smaller changes too.

For internal communication, the team uses Slack (interestingly – despite the "eat your own dog food" saying at Microsoft – not Microsoft Teams). Most other informational resources and communication are available at GitHub. Some data on GitHub, like the user documentation that is in a separate repository, is deployed to the respective Microsoft documentation pages.

Development Cycle

Gamma and his team follow an agile system that he adapted from his previous experience in his work at the Eclipse project (Gamma, 2005). Development happens in iterations that roughly start with every month, but always beginning on a Monday and ending on a Friday. In the first week, the iteration is planned. Issues that will be done in the iteration are selected, labeled as `plan item`, and split into parts that need to be done to finish the issue. Also, bugs get a priority. Weeks 2 and 3, sometimes longer, are used to implement the plan items. The final week of an iteration is called endgame. In the endgame, no new feature contributions are allowed, the week is reserved to only test and fix bugs. The community plays an active part in the endgame: At the beginning of the endgame, the team releases a so-called insider build. Community members can use that version and report bugs. Each End Game is coordinated by another person, the “endgame master”, who selects an “endgame buddy” from the other team (Zurich/Redmond) to help him. At the end of the endgame, a stable build is released. Ideally, it then contains no bugs anymore (Development Process, 2020).

Iterations often focus on a common goal; there are also some special types of iterations. For example, at the time of writing, the current iteration one of the annual “House Keeping Iterations”, in which solely serve the purpose of organizing the huge amount of issues that were created over the year (Issue Grooming, 2019).

The biggest change that Gamma made with his team to this process in comparison to the one in his previous project is the length of the iterations: for Eclipse, they had a length of 6 weeks (Gamma, 2005). That change is interesting, because in 2005 he argued shorter iterations would be bad since constantly starting and stopping development would slow everything down. There is not much official information on why he decided to shorten the iterations, possible reasons include the type of project: Visual Studio Code is much simpler, is to some extent more modular, and has less features than an IDE like Eclipse. That makes the development workflow simpler, less time might necessary to achieve similar things.

Contributing

There are many ways one can contribute to Visual Studio Code, as listed in the `README.md` file. The most popular and easy way is to report bus and suggest feature requests via the issue tracker, around 50 to 250 people use that opportunity daily (Issue Grooming, 2019). The repository offers issue template for those cases and streamlines the process using bots for automated issue categorization/labeling.

Another way one can contribute without much technical knowledge, is in the documentation, which contributors can correct or translate. There was a separate platform for this, but since March 2016 Documentation for Microsoft products is available via a separate GitHub repository.

Lastly, one can contribute code. Because this is the most technical way of contribution, the project wiki contains an extensive page explaining all the steps, tools, and knowledge that is necessary to contribute, such as how compiling and debugging works and which requirements to linting there are.

When contributing code or documentation to a Microsoft Project such as Visual Studio Code for the first time, the contributor must agree to Microsoft’s Contribution License Agreement (CLA). A GitHub bot leads the first-time contributor through the process. The contributor mainly declares that he has all rights to the changes he made, and that Microsoft can use them. The document contains a form in which not only a signature but also name and address must be filled in.

The usual way to start a contribution is by searching a fitting issue. There are many issues labeled `help wanted`, and especially for beginners, some labeled `good first issue`. Usually, the issues describe the problem but don’t describe technical details. The contributor then comments on the issue and signals his interest in fixing it, often asking the assignee, a member of the core team, for details where the root of the problem can be found. When the assignee accepts the help, the contributor can start working on the issue in an own fork. The original issue needs to be referenced in the pull request. The assignee can then review the changes, and in case it is necessary either ask for additional changes or add changes on his own. In case the pull request doesn't adhere to quality standards, it can be rejected too. A common reason for rejection is that the implementation is not lightweight enough (How to Contribute, 2020).

Conclusion & Look Back

By various metrics, there are few Open Source projects that are as big as Visual Studio Code. Even though its size and the big amount of available resources of the project might seem a bit intimidating at first, the Visual Studio Code community nevertheless presents itself as welcoming to new contributors, especially for smaller contributions. Digging into the depths of an Open Source project didn't just help me to extract the data for this report, while researching I also came across some interesting stories, that didn't fit in here about the world of Open Source – such as one of a Microsoft ex-employee who decided to quit his work on Visual Studio Code to live a nomadic life, work on his own projects and finance himself via donations. Or the story of how Bill Gates and his friends developed Microsoft's first product without access to the processor that it ran on, in Assembly – and still managed to make it run perfectly. Before starting this report, I feared that it might be hard to find a possibility to contribute to this project. Thankfully, I found out that despite being led by an established corporation, getting into contributing is not hard and surprisingly not too bureaucratic.

References

Ajani, S, sana.ajani@microsoft.com

Contributors. (2020). Retrieved from GitHub: https://github.com/microsoft/vscode/graphs/contributors

Developer Survey Results 2019. (2019). Retrieved from Stack Overflow: https://insights.stackoverflow.com/survey/2019

Development Process. (2020, 8 26). Retrieved from Github: Visual Studio Code Wiki: https://github.com/microsoft/vscode/wiki/Development-Process

Dias, C., & Ajani, S. (2020, 5 25). *Ask the Team: Visual Studio Code I COM01.* Retrieved from YouTube: https://www.youtube.com/watch?v=VRiaDG--A4k

Dias, C., Juarez, S., & Gamma, E. (2015, 4 28). *Meet the Visual Studio Code team.* Retrieved from Channel 9: https://channel9.msdn.com/events/Build/2015/C9-05

Foley, M. (2018, 7 7). *Microsoft's 'future CEO of GitHub' speaks out on Atom, keeping GitHub independent and more*. Retrieved from ZDNet: https://www.zdnet.com/article/microsofts-future-ceo-of-github-speaks-out-on-atom-keeping-github-independent-and-more/

Gamma, E. (2005, 6 28). Eclipse's Culture of Shipping: A Conversation with Erich Gamma, Part V. (B. Venners, Interviewer) Retrieved from https://www.artima.com/lejava/articles/eclipse_culture.html

Gates, W. (n.d.). Transcript of a Video History Interview with Mr. William "Bill" Gates. (D. Allison, Interviewer) Retrieved from https://americanhistory.si.edu/comphist/gates.htm#tc11

Greene, T. C. (2001, 06 2). *Ballmer: 'Linux is a cancer': Contaminates all other software with Hippie GPL rubbish*. Retrieved from The Register: https://www.theregister.com/2001/06/02/ballmer_linux_is_a_cancer/

Haverbeke, M. (2020). *Release notes and version history*. Retrieved from CodeMirror: https://codemirror.net/doc/releases.html

How to Contribute. (2020, 11 28). Retrieved from GitHub: Visual Studio Wiki: https://github.com/microsoft/vscode/wiki/How-to-Contribute

Issue Grooming. (2019, 10 15). Retrieved from GitHub: Visual Studio Code Wiki: https://github.com/microsoft/vscode/wiki/Issue-Grooming

Maetzel, K. (2020, 10 11). *Issue: 2020 Issue Grooming - review, categorization, assignment #108472*. Retrieved from GitHub: https://github.com/microsoft/vscode/issues/108472

Microsoft acquires GitHub. (2018, 6 4). Retrieved from About Microsoft: https://news.microsoft.com/announcement/microsoft-acquires-github/?return=https%3A%2F%2Fnews.microsoft.com%2Fabout%2F

MICROSOFT SOFTWARE LICENSE TERMS: MICROSOFT VISUAL STUDIO CODE. (2020). Retrieved from Visual Studio: https://code.visualstudio.com/License/

Paoli, J. (2014, 4 3). *.NET Foundation Established to Foster Open Development*. Retrieved from Microsoft Open Technologies: https://web.archive.org/web/20150521154900/https://msopentech.com/blog/2014/04/03/net-foundation-established-foster-open-development/

The State of the Octoverse. (2020). Retrieved from GitHub: https://octoverse.github.com/

Thurrot, P. (1997, 04 17). *Microsoft sets pricing on Visual Studio 97*. Retrieved from itprotoday: https://www.itprotoday.com/windows-78/microsoft-sets-pricing-visual-studio-97

Visual Studio 2019 Product Family System Requirements. (2019, 2 4). Retrieved from Microsoft Docs: https://docs.microsoft.com/en-us/visualstudio/releases/2019/system-requirements

Warren, T. (2020, 5 18). *Microsoft: we were wrong about open source: Microsoft has embraced open source and even Linux in recent years*. Retrieved from The Verge: https://www.theverge.com/2020/5/18/21262103/microsoft-open-source-linux-history-wrong-statement