EINFÜHRUNG IN GITHUB

Niklas Tyrakowski

31. AUGUST 2021

UNIVERSITÄT HAMBURG
Fakultät für Mathematik, Informatik und Naturwissenschaften
Veranstaltung: Proseminar Softwareentwicklung in der Wissenschaft
Betreuer: Jannek Squar

0 Gliederung

1 Einleitung – Was ist GitHub?

GitHub ist ein netzbasierter Dienst zur Versionsverwaltung für Software-Entwicklungsprojekte[1], welcher auf dem Namensgeber Git basiert. Mit diesem Dienst haben 65 Millionen Nutzer bereits über 200 Millionen Repositories erstellt[2]. Ein Repository (Englisch für Ablage oder Lager) ist ein Ort zur Speicherung und Beschreibung von digitalen Objekten[3], hier vor allem Software-Repositories.

Hinter dem Dienst GitHub, steht die GitHub Inc. unter der Führung von CEO Nat Friedman. Auch wenn GitHub 2018 für 7,5 Milliarden US-Dollar an Microsoft verkauft wurde[4], so arbeitet sie, laut Microsoft, noch als unabhängiges Unternehmen.

2 Motivation

Da es in der heutigen Zeit fast undenkbar ist Software nicht in Teams zu entwickeln, werden Dienste wie GitHub immer beliebter. Sie vereinfachen es, gemeinsam an Projekten zu arbeiten, das Projekt in Stand zu halten, Fehler zu erkennen und auch die Auslieferung wird simpler gestaltet.

Insbesondere auch in der Zukunft wird es immer wichtiger sein, sich in einem Team über eine zentrale Plattform austauschen zu können, nicht nur zur Planung, sondern auch zur Durchführung und Entwicklung von Software-Programmen. Das wurde speziell im vergangenen Jahr durch die Corona-Pandemie und die daraus steigenden Home-Office Zahlen deutlich.
Durch GitHub war es Entwicklern möglich ihre Änderungen am Code mit anderen zu teilen und diesen dadurch zu verbessern.

Ziel dieser Hausarbeit ist es den Prozess der Zusammenarbeit mit GitHub näher zu beleuchten und zu erläutern.

[1] (1) Wikipedia
[2] (6) GitHub About
[3] (16) Wikipedia Repository
[4] (7) dpa

3 Historie[5]

GitHub Incorporated wurde 2007 in San Francisco gegründet und im Februar des Folgejahres wurde mit dem Projekt GitHub angefangen. Bereits zwei Monate später, im April 2008, ging die Domain „github.com“ online.

Der erste große Erfolg wurde im Jahr 2011 errungen, als GitHub, gemessen an Commits, der populärste Dienst seiner Art war. „Ein Commit ist ein Ausdruck aus der Softwaretechnik, der die bestätigende Freischaltung einer oder mehrerer Änderungen beschreibt“[6].

Daraufhin investierte Andreessen Horowitz im Juli 2012 einhundert Millionen US-Dollar, drei Jahre später kamen noch einmal 250 Millionen Dollar von diversen Fonds, hinzu. Zu diesem Zeitpunkt hatte GitHub bereits zehn Millionen Nutzer und 26 Millionen Repositories.

Nicht nur als Dienst wurde GitHub im Laufe der Jahre immer populärer, sondern auch in der Wissenschaft. Denn im Oktober 2016 wurde GitHub bei einem Prozent aller Publikationen aus der Informatik als Quelle angegeben.

Nachdem dann im Juni 2018 der Verkauft an Microsoft vollzogen wurde, übernahm die GitHub Inc. selbst einige Unternehmen. Noch im selben Jahr Spectrum[7], ein Dienst der die Kommunikation über die Plattform vereinfacht hat. Im Jahr darauf waren es sogar drei Dienste die übernommen wurde, Dependabot[8], Pull Panda[9] und Semmle.[10] Dependabot ist ein Tool, mit dem Abhängigkeiten (engl. Dependencies) in einem Projekt durchsucht und automatisch aktualisiert werden können. Pull Panda macht es einfacher, neuen Code mit bestehendem Code zu vergleichen und anschließend zusammenzuführen. Und mit Semmle kann Sourcecode automatisch auf Sicherheitslücken überprüft werden.

GitHub kann nicht nur auf einem Desktop Computer benutzt werden, sondern auch auf dem Smartphone können Nutzer seit März 2020, über eine eigens entwickelte App, ihre Repositories verwalten.

[5] (1) Wikipedia GitHub
[6] (17) Wikipedia Commit
[7] (18) Savia Lobo
[8] (19) Stergios Georgopoulos
[9] (20) Ravie Lashmanan
[10] (21) Frederic Lardinois

4 Features[11]

Der zentrale Aspekt bei GitHub ist gemeinsames Programmieren. Dieser wird zum Beispiel durch Codespaces ermöglicht. In diesem ist es den Nutzern möglich, Code zu schreiben, testen, debuggen und schließlich auch auszuliefern.

Des Weiteren werden an Teilhaber und Abonnenten des Projektes stets Benachrichtigungen gesendet, wenn es Änderungen im Projekt gibt. Das passiert zum einen, wenn ein neuer Pull Request erstellt wird, mit welchem, Änderungen eines Nutzers in das laufende Projekt gezogen werden sollen. Dieser wird dann von anderen Nutzern überprüft, hiermit kann ein gewissen Standard im Sourcecode garantiert werden.

Zu jedem Repository gibt es auch ein eigenes Forum, in dem sich die Entwickler austauschen können, das ist sehr nützlich, wenn mehrere von ihnen an einem Bug oder Feature zusammenarbeiten.

Zum einen kann in privaten Organisationen auf GitHub gearbeitet werden. Andererseits gibt es auch die Möglichkeit ein öffentliches Projekt zu erstellen, an dem alle Nutzer der Plattform teilhaben können. Das sind dann die sogenannten Open Source Projekte, hiervon gibt es sehr viele. Die Corona Warn App wird zum Beispiel über GitHub entwickelt, so können Datenlücken und Programmfehler sehr schnell erkannt und ausgebessert werden.

Auch die Automatisierung von Arbeitsabläufen ist möglich. Das wird größtenteils durch GitHub Actions, was später noch einmal im Detail untersucht wird, erreicht. Im Grunde ist es möglich, jegliche administrative Aufgabe automatisch abhandeln zu lassen. Das fängt beim Zuweisen von Aufgaben auf die Entwickler an und endet beim letztlichen Ausliefern der fertigen Software.

Schon seit längerer Zeit stehen Datenschutz und Datensicherheit im Fokus der IT-Branche. Hier gibt es die zwei Faktor Authentifikation, mit der sich Benutzer gegen Angriffe auf die Nutzerkonten schützten können. Des Weiteren gibt es Codes mit denen Kontos zurückgeholt werden können, falls diese gelöscht oder gehackt werden sollten. Außerdem kann eine Telefonnummer hinterlegt um noch eine weiter Sicherheitsschicht anzulegen.

Nicht nur bei den Nutzern wird auf Sicherheit gesetzt, sondern auch im Code selbst. Das wird unter anderem automatisch durch Dependabot und Semmle erreicht. Darüber hinaus kann die Überprüfung des Codes durch andere vorgeschrieben werden. Hier kann auch die Anzahl der Entwickler festgelegt werden, welche sich die Änderungen anschauen und absegnen sollen.

Weiterhin gibt es auch die Möglichkeit einfach ein privates Repository anzulegen, welches dann mit ausgewählten Nutzern geteilt wird. Hier kann der Benutzer also selber entscheiden, wer in einem Projekt mitarbeiten kann und wer nicht. Diese privaten Projekte sind ausschließlich für eingeladene Nutzer einsehbar.

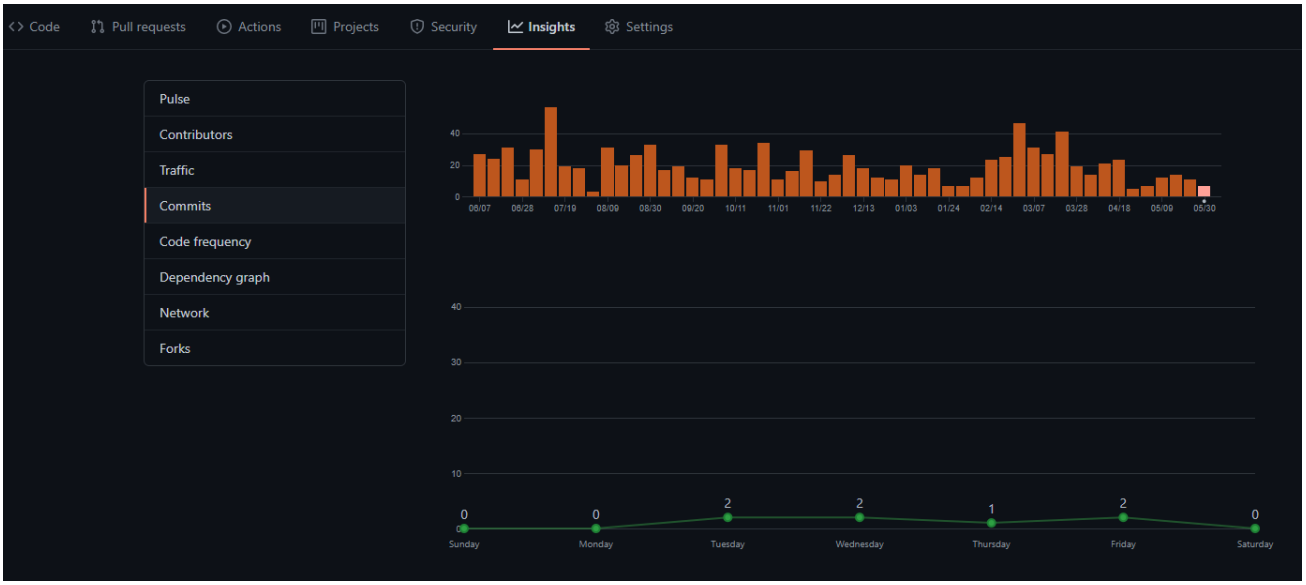
[11] (22) GitHub Features

Das Management eines Repositories kann zum einem über einen Browser getätigt werden, als auch über eine App, für Desktop und Smartphone. Dadurch ist also auch von Unterwegs sehr gut möglich, alle Aktivitäten zu verfolgen.

Hier steht einem der Issues Tab zur Verfügung, in dem Nutzer Fehler oder Anforderungen erstellen können. Das funktioniert ähnlich wie in einem Forum, unter jedem Beitrag können Antworten abgegeben werden. Hierdurch wird eine Diskussion angeregt und der Code wir indirekt besser gemacht.

Zusätzlich zu dem Issues Tab, gibt es auch Labels die an jeden Beitrag gepinnt werden können. Diese können selbst erstellt werden. Das größte Nutzen diese Labels ist die Organisation von Themen, hier kann zum Beispiel ein Label „Bug“ erstellt werden, mit dem dann alle Fehler versehen werden. Die Möglichkeiten sind hier unbegrenzt und jeder kann selber entscheiden, wie er es handhaben möchte.

In dem Insights Tab stehen Statistiken zum Repository, unter anderem können die Anzahl der Commits oder die Anzahl der Entwickler eingesehen werden.



Die Zusammenarbeit mit vielen Nutzern wird durch Organisationen vereinfacht, denn hier können verschiedene Teams, mit unterschiedlichen Aufgaben und Berechtigungen, erstellt werden. So kann eine Unternehmens-Hierarchie auch in einem Projekt abgebildet werden.

Organisationen in GitHub sind privat. Das heißt, dass sie nicht von alleine betreten werden können, sondern es bedarf einer Einladung, bevor ein Nutzer sie betreten kann.

Bei Open Source Projekten gibt es die Möglichkeit Spenden zu erhalten, denn an solchen Projekten gibt es einen Sponsor Button. Hier können Nutzer die Projekte finanzieren die sie benötigen um ihre eigene Software zu bauen oder einfach als Unterstützung spenden.

Weiterhin kann GitHub auch als Lernplattform benutzt werden. Es können also Nutzer die wenig oder keine Erfahrung mir Sourcecode haben auch lernen. Dieser Unterdienst heißt Learning Lab und hier können erfahrene Nutzer Aufgaben stellen, die einem Anfänger beibringen können Software zu entwickeln. Darüber hinaus gibt es auch Aufgaben für fortgeschrittene Nutzer, es kann sich also auch weitergebildet werden.

5 Zusammenarbeit in GitHub

Für die Zusammenarbeit gibt es unterschiedliche Wege. Dies kann bereits beim Erstellen des Repositories festgelegt werden. Zum einen kann ein privates Projekt eröffnet und nur mit eingeladenen Leuten zusammenarbeiten werden, die andere Option wäre ein öffentliches Repository. Auch hier gibt es dann zwei Optionen, auf der einen Seite die Open Source Repositories bei denen alle Nutzer gleichermaßen teilhaben können, dem gegenüber stehen die strukturierten öffentlichen Repositories, welche durch Organisationen verwaltet werden und in denen jede Unterorganisation verschiedene Pflichten und Rechte hat.

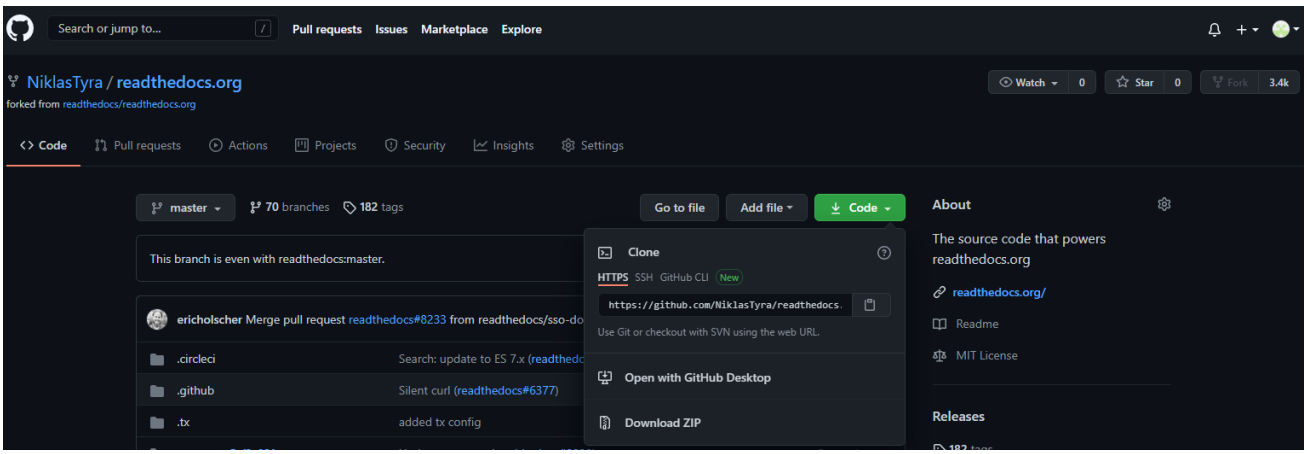
Des Weiteren gibt es auch die Möglichkeit an Projekten zu arbeiten, ohne Code schreiben zu müssen. Dies geschieht über den „Issues Tab“, bei dem an Diskussionen teilgenommen und Vorschläge geäußert oder diskutiert werden können.

Wahlweise kann auch unter den „Pull Requests“ Code kontrolliert werden um einen gewissen Code Standard zu garantieren. Das ist eine sehr gute Möglichkeit andere Code Stile zu sehen und so auch seine eigenen Fähigkeiten als Entwickler zu verbessern.

Wer allerdings lieber Code schreibt und an einem bestehenden Projekt teilnehmen möchte, für den schauen wir uns den Prozess einmal genauer an.

Dieser beginnt mit dem „forken“ eines bestehenden Repositories. Eine Fork (Englisch für Gabel oder Abspaltung), wird in der Softwareentwicklung als Begriff für einen Entwicklungszweig benutzt.[12]

Nach dem erfolgreichen Anhängen des Repositories an den eigenen Account, muss man sich im Folgeschritt den Sourcecode duplizieren, was mit dem „clonen“ erreicht wird. Hierbei gibt es verschiedene Möglichkeiten. Zum einen das Herunterladen einer ZIP-Datei, in welcher der Programmordner enthalten ist, der auf dem lokalen PC entpackt werden muss und anschließend mit einer Entwicklungsumgebung geöffnet wird. Die nächste Möglichkeit ist das Öffnen des Projektes in der GitHub App. Das ist allerdings nur auf dem Desktop und nicht auf dem Smartphone möglich. Die letzte Möglichkeit ist das Importieren der Projektstruktur direkt in die Entwicklungsumgebung, über einen Link.

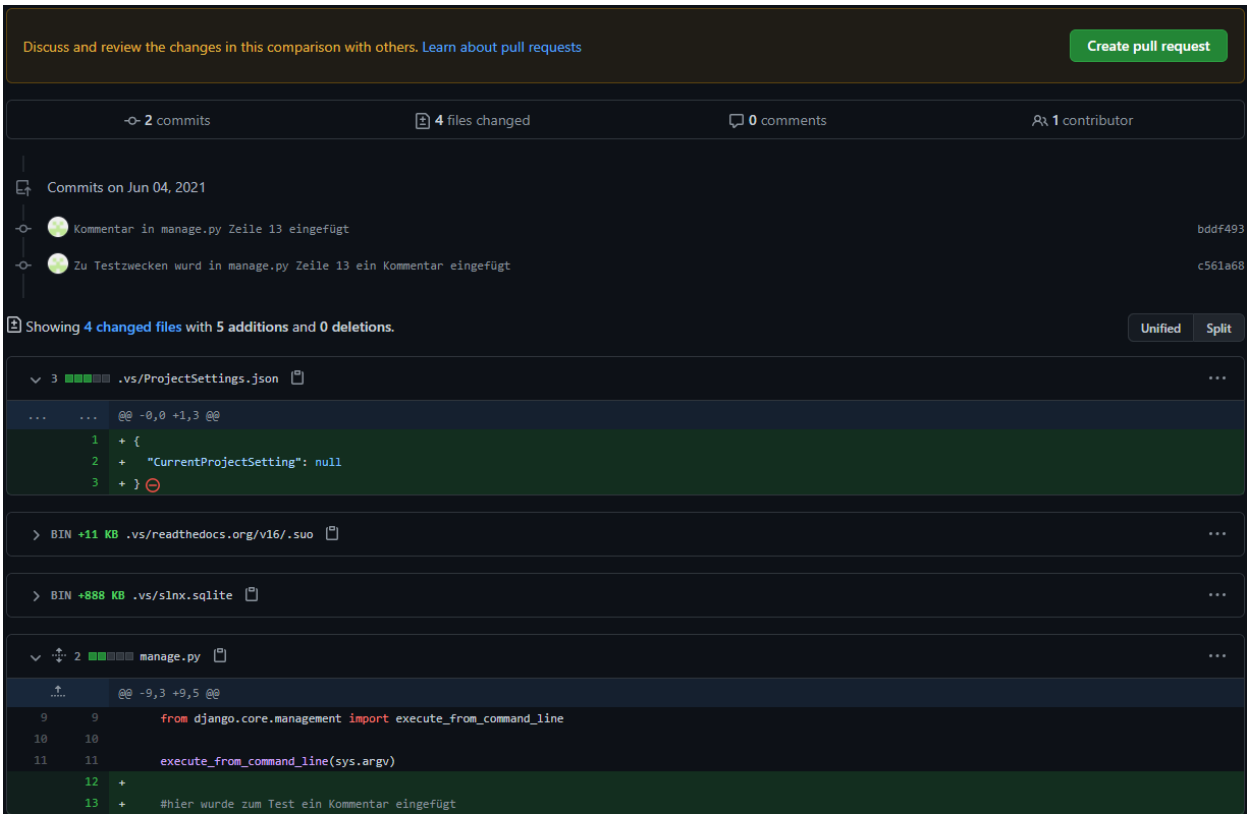


[12] (23) Wikipedia Abspaltung

Der nächste Schritt ist, das Auschecken eines neuen „Branches“. „Ein Branch, zu Deutsch Zweig, ist eine Verzweigung zu einer neuen Version, so dass unterschiedliche Versionen parallel im selben Projekt weiterentwickelt werden können.“[13] Hiermit wird sichergestellt, dass die Änderungen im Code später nur dann übernommen werden, wenn diese auch kompatibel mit der Ursprungsversion sein.

Nun kommt der wichtigste Schritt, das Bearbeiten des Codes. Hier werden nun die gewünschten Änderungen vorgenommen und gegebenenfalls Code ausgebessert. Das passiert entweder direkt in der GitHub App oder in der bevorzugten Entwicklungsumgebung des Nutzers.

Nachdem die geänderte Version hochgeladen wurde, muss diese von anderen Nutzern kontrolliert werden, bevor sie dann in das laufende Programm integriert werden kann. Das passiert, indem ein „Pull Request“ erstellt wird.



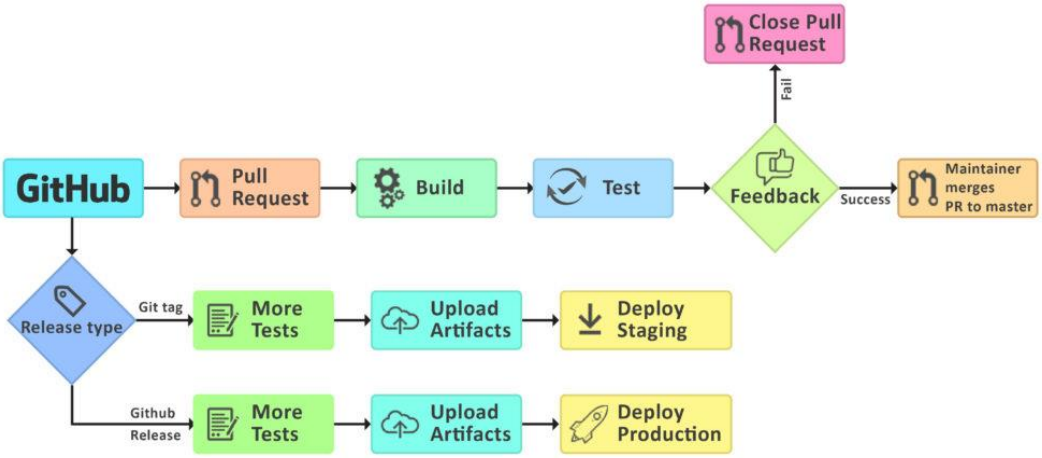
In jedem Pull Request werden die vorgenommenen Änderungen markiert, sodass die Kontrolleure diese einfacher finden können. Wenn bei der Überprüfung, Fehler oder Verbesserungsvorschläge aufkommen, müssen diese umgehend vom Entwickler vorgenommen werden.

Im Anschluss kann der Branch in den Hauptbranch, welcher meist Master oder Main heißt, „gemerged“ werden. Merge (aus dem Englischen, verschmelzen oder zusammenführen) ist ein Vorgang, in dem zunächst zwei Versionen verglichen und anschließend konfliktfrei zusammengeführt werden.[14]

[13] (24) Wikipedia Versionsverwaltung
[14] (25) Wikipedia Merge

6 GitHub Actions

In einem großen Projekt fallen viele organisatorischen Aufgaben an, um also einen möglichst effizienten und reibungslosen Arbeitsprozess zu schaffen führte GitHub Automatisierungen ein. Es ist möglich alle algorithmischen Abläufe zu automatisieren. Dies wird auch „continuous integration and continuous delivery“ genannt, denn es kann ständig gearbeitet werden, wenn die organisatorischen Abläufe automatisiert sind.



[15]

CI/CD Workflow

Dieser Prozess kann in den „Actions“ eingerichtet werden. Speziell werden hier „Events“ deklariert, welche dann von interner Logik abgehandelt werden können, dies reicht vom Zuweisen von Fehlern an Entwickler bis hin zum Ausliefern der Software an Kunden.

Wir wollen uns Ersteres genauer anschauen. Der erste Schritt, nachdem ein Fehler gemeldet wurde, ist die Beschriftung. Hier wird zunächst festgestellt was die Art des Fehlers ist, zum Beispiel kosmetisch oder sogar fatal. Letztere müssen in der Priorisierung natürlich weiter oben stehen.

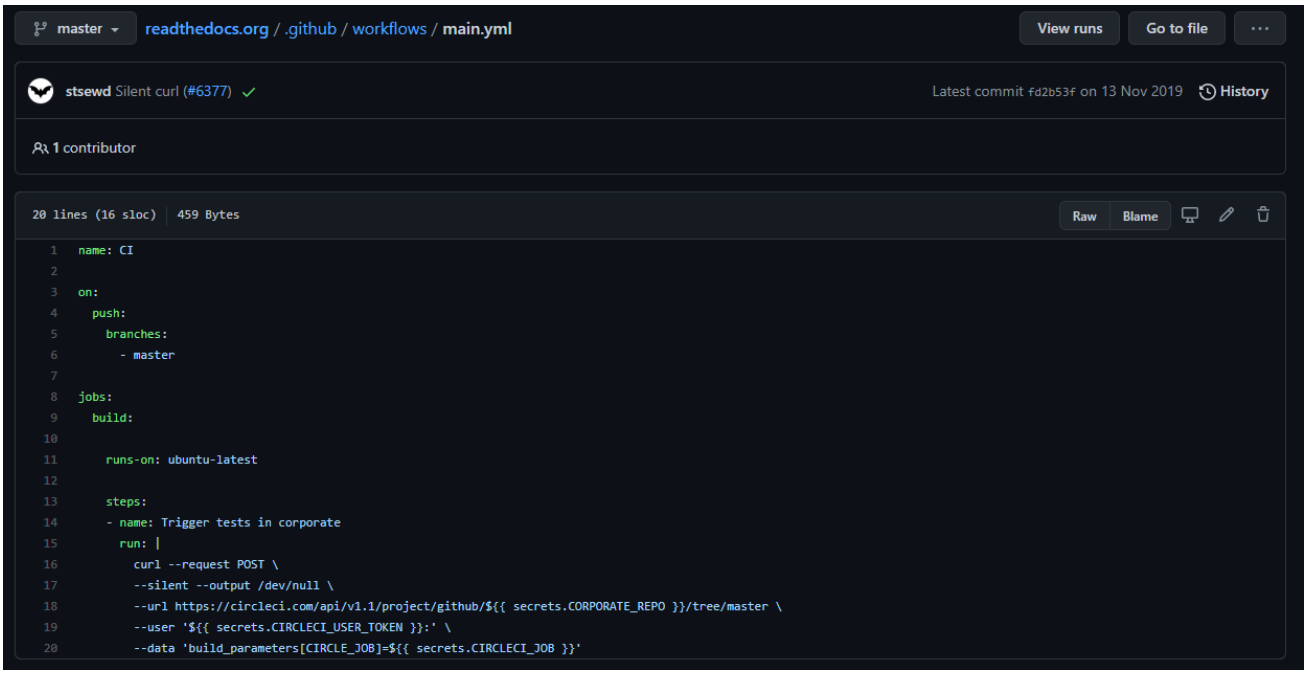
Auch das Sortieren kann anschließend von GitHub übernommen werden. Das Festlegen der Reihenfolge steht jedem Projektmanager selber zu. Diese wird dann übernommen, sodass zum Beispiel fatale Fehler oben und kosmetische ganz unten stehen.

Nachdem alle Fehler beschriftet und sortiert sind, wäre es von Vorteil, wenn sich die Entwickler direkt an die Ausbesserung geben würden. In der Realität passiert das meist nur, wenn die Aufgaben auch zugewiesen werden. Ebenso wie die bisherigen Schritte, kann auch dieser automatisiert werden. So können die Probleme nach Schwierigkeit verschiedenen Entwicklern zugewiesen werden. Das wird alles in den „Workflows“ dokumentiert und festgelegt.

[15] (13) Biarca

Diese Arbeitsabläufe sind in YAML geschrieben, „YAML ist eine vereinfachte Auszeichnungssprache (englisch markup language)“[16] YAML ist im Gegensatz zu XML oder HTML deutlich einfacher von Menschen zu lesen, da sie nur mit assoziativen Listen, Arrays und Skalaren arbeitet.[17]

Auf GitHub gibt es bereits sehr viele Vorlagen zu verschiedensten Abläufen, sodass nicht jeder Schritt selbst automatisiert werden muss. Diese sind im Actions Tab des Projektes zu finden. Bereits eingerichtete Workflows sind alle im selben Ordner hinterlegt, dieser ist unter diesem Pfad hinterlegt: „…/.github/workflows/<workflow>.yml“.



Der dargestellte Ablauf, bezieht sich auf einen Push zum Masterbranch, dies erkennt man am Stichwort „on“. Wenn nun also etwas in den Masterbranch gepusht wird, so muss GitHub das Programm neu erstellen, denn unter „jobs“ ist „build“ festgelegt. Es ist hier sehr gut zu erkennen, wie einfach es ist YAML zu lesen und zu verstehen.

Natürlich besteht auch die Möglichkeit jeden Arbeitsablauf selber zu schreiben. Dazu muss nur eine neue YAML Datei, in welcher der gewünschte Ablauft deklariert ist, hinterlegt werden.

[16] (26) Wikipedia YAML
[17] (26)

7 Fazit

Alles in Allem ist GitHub also ein sehr gutes Tool für die Zusammenarbeit in Softwareprojekten. Hinzukommt, dass es der größte Cloudsoftware Dienst der Welt, mit Microsoft im Hintergrund ist, was gewährleistet, dass es noch über Jahre bestehen und weiterentwickelt wird.

Weiterhin ist GitHub vielseitig einsetzbar, ob alleine oder in Teams gearbeitet wird. Für alles gibt es adäquate Lösungen. Allerdings kann nicht nur gearbeitet werden, sondern auch zur Weiterbildung gibt es das Learning Lab. Darüber hinaus können sich die Nutzer in ihren Profilen präsentieren und sich so auch bei größeren Firmen wie Microsoft empfehlen.

Für Nutzer die nicht gerne oder noch nicht so gut Code selber schreiben, besteht auch die Möglichkeit sich anders an Projekten zu beteiligen. Zum einen durch die Teilnahme an Diskussionen und das Gegenlesen von Pull Requests, zum anderen durch Testen und das Protokollieren von Fehlern.

Darüber hinaus ist durch den Namensgeber Git auch eine Versionskontrolle eingebaut. Hierdurch wird stets funktionierender Code sichergestellt

Außerdem werden durch GitHub Actions die administrativen Aufgaben in Projekten fast komplett automatisiert, was Zeit spart und den Fokus auf die wichtigeren Abläufe in Projekten richtet.

Zusätzlich gibt es in vielen Entwicklungsumgebungen bereits ein Git/GitHub Add-in, mit dem man über eine gewohnte Umgebung mühelos mit GitHub arbeiten kann.

Generell bleibt also festzuhalten, dass die Zukunft der Entwicklung fast ausschließlich über Cloudplattformen stattfinden wird. Hier gibt es natürlich mehrere Optionen, denn neben GitHub bietet GitLab ähnliche Möglichkeiten, aber auch AzureDevOps, welches auch von Microsoft betrieben wird. DevOps wird allerdings nicht mehr weiterentwickelt.

8 Literaturverzeichnis

1. Wikipedia. Github. [Online] 04. 03 2021. [Zitat vom: 02. 06 2021.] https://de.wikipedia.org/wiki/GitHub.

2. GitHub, Inc. About continious integration. [Online] 2021. [Zitat vom: 28. 05 2021.] https://docs.github.com/en/actions/guides/about-continuous-integration.

3. German et al., Daniel M. *The Promises and Perils of Mining GitHub.* Hyderbad, Indiana USA : University of Victoria, 2007.

4. Stückler, Moritz. Was ist eigentlich Github? [Online] t3n digital pioneers, 24. 02 2020. [Zitat vom: 26. 05 2021.] https://t3n.de/news/eigentlich-github-472886/.

5. Holland, Martin. Microsoft kauft GitHub für 7,5 Milliarden US-Dollar. [Online] heise, 04. 06 2018. [Zitat vom: 26. 05 2021.] https://www.heise.de/newsticker/meldung/Microsoft-kauft-GitHub-fuer-7-5-Milliarden-US-Dollar-4067633.html.

6. Github Inc. Github About. [Online] 2021. [Zitat vom: 26. 05 2021.] https://github.com/about.

7. dpa. EU-Kommission: Übernahme von GitHub durch Microsoft genehmigt. [Online] 20. 10 2018. [Zitat vom: 26. 05 2021.] https://www.heise.de/newsticker/meldung/EU-Kommission-Uebernahme-von-GitHub-durch-Microsoft-genehmigt-4197599.html.

8. Cleeren, Gill. *Github: Getting Started.* [Videokurs] s.l. : pluralsight, pluralsight, 2020.

9. Bell, Peter. *Code School: Mastering GitHub.* [Videokurs] s.l. : pluralsight, 2014.

10. GitHub. YouTube. *GitHub Actions - Now with built-in CI/CD! Live from GitHub HQ.* [Online] GitHub, 08. 08 2019. [Zitat vom: 04. 06 2021.] https://www.youtube.com/watch?v=E1OunoCyuhY.

11. readthedocs.org. GitHub Repository. [Online] readthedocs.org, 2021. [Zitat vom: 04. 06 2021.] https://github.com/readthedocs/readthedocs.org.

12. Cutrell, Jonathan. How to Collaborate On GitHub. [Online] 22. 08 2013. [Zitat vom: 04. 06 2021.] https://code.tutsplus.com/tutorials/how-to-collaborate-on-github--net-34267.

13. Biarca. Biarca.io. [Online] 13. 02 2019. [Zitat vom: 04. 06 2021.] https://biarca.io/2019/02/biarca-helps-linux-foundation-develop-and-deploy-a-ci-cd-workflow-for-their-certification-platform/.

14. GitHub. GitHub Actions. [Online] 2021. [Zitat vom: 04. 06 2021.] https://docs.github.com/en/actions/learn-github-actions/introduction-to-github-actions.

15. Sacolick, Isaac. InfoWorld. [Online] 17. 01 2020. [Zitat vom: 07. 06 2021.] https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html.

16. Wikipedia. Repository. [Online] 14. 03 2021. [Zitat vom: 17. 08 2021.] https://de.wikipedia.org/wiki/Repository.

17. —. Commit. [Online] Wikipedia, 21. 09 2018. [Zitat vom: 17. 08 2021.] https://de.wikipedia.org/wiki/Commit.

18. Lobo, Savia. GitHub acquires Spectrum. [Online] Packt, 03. 12 2018. [Zitat vom: 2021. 08 17.] https://hub.packtpub.com/github-acquires-spectrum-a-community-centric-conversational-platform/.

19. Georgopoulos, Stergios. GitHub acquires Dependabot. [Online] Neowin, 23. 05 2019. [Zitat vom: 17. 08 2021.] https://www.neowin.net/news/github-acquires-dependabot-launches-github-sponsors/.

20. Lakshmanan, Ravie. GitHub acquires Pull Panda. [Online] thenextweb, 19. 06 2019. [Zitat vom: 17. 08 2021.] https://thenextweb.com/news/github-acquires-pull-panda-and-makes-its-code-review-tools-available-for-free.

21. Lardinois, Frederic. GitHub acquires code analysis tool Semmle. [Online] techcrunch, 18. 09 2019. [Zitat vom: 17. 08 2021.] https://techcrunch.com/2019/09/18/github-acquires-code-analysis-tool-semmle/.

22. GitHub. GitHub Features. [Online] GitHub, 2021. [Zitat vom: 17. 08 2021.] https://github.com/features.

23. Wikipedia. Abspaltung (Softwareentwicklung). [Online] Wikipeida, 28. 03 2021. [Zitat vom: 18. 08 2021.] https://de.wikipedia.org/wiki/Abspaltung_(Softwareentwicklung).

24. —. Versionsverwaltung. [Online] Wikipedia, 09. 06 2021. [Zitat vom: 18. 08 2021.] https://de.wikipedia.org/wiki/Versionsverwaltung.

25. —. Merge. [Online] Wikipedia, 17. 03 2021. [Zitat vom: 18. 08 2021.] https://de.wikipedia.org/wiki/Merge.

26. —. YAML. [Online] Wikipedia, 16. 08 2021. [Zitat vom: 19. 08 2021.] https://de.wikipedia.org/wiki/YAML.