

1 Einleitung

In der IT ist Open Source mittlerweile ein fester Bestandteil der gesamten Infrastruktur. Mehr als die Hälfte aller Web Server laufen unter Open-Source-Lizenzen [W3T22]. Die meistgenutzten Frontend Frameworks sind ebenfalls alle Open Source. [Sta]

Der Begriff Open Source ist den meisten Softwareentwicklern wahrscheinlich bekannt, aber was genau steckt dahinter? Die Antwort ist weitaus mehr als *nur* quelloffener Code und kostenlose Software. Die *Open Source Initiative* hat eine klare Definition für Open Source. Wie der Name schon sagt, muss der Quellcode offen liegen, des Weiteren gelten allerdings auch Voraussetzungen, wie beispielsweise, dass Nutzer den Quellcode verändern und weitergeben dürfen [Ope].

Mit dieser Arbeit soll, basierend auf ausgewählten Open-Source-Projekten und einer Umfrage, herausgefunden werden welche Faktoren entscheidend zum Erfolg eines Projektes beitragen. Hierbei wird hauptsächlich von der Nutzerperspektive ausgegangen, wobei mit Nutzer nicht nur die Endnutzer der Software, sondern auch Softwareentwickler gemeint sind, die Open Source Produkte wie Bibliotheken etc. in eigenen Projekten einbauen.

Ein zentraler Punkt dieser Ausarbeitung sind die extrinsischen sowie intrinsischen Anreize, die Nutzer zur Auswahl eines Produktes motivieren [Mid12]. Aspekte wie die interne Führung und Organisation der Projekte wird hierbei nicht thematisiert.

1.1 Erfolg definieren

In der Literatur wird häufig in verschiedene Bereichen des Erfolgs unterteilt. Im Artikel von Midha und Palvia wird zwischen *Markterfolg* und *technischem Erfolg* unterschieden. Markterfolg definiert Midha et al. als Grad des Nutzerinteresses an ein Projekt, welches sich in der Beliebtheit des Projektes widerspiegelt. Den technischen Erfolg definiert Midha et al. durch die Entwickleraktivität, d.h. durch den Aufwand, den die Entwickler für das Projekt betreiben. Beispielsweise die Häufigkeit und Frequenz von Updates und neuen Versionen. [Mid12].

In Steward et al. wird zwischen *Nutzerinteresse* und *Entwickleraktivität* unterschieden [Ste06]. Subramaniam et al. geht sogar weiter und unterteilt in *Nutzerinteresse*, *Entwicklerinteresse* und *Projekttätigkeit* [Sub09].

Diese Bereiche werden getrennt betrachtet, da verschiedene Faktoren unterschiedlichen Einfluss auf den Erfolg eines Projektes haben. Während sich wachsendes Interesse bei Nutzern positiv auf den Markterfolg auswirkt, wirkt sich die Entwickleraktivität positiv den technischen Erfolg aus [Mid12, Ste06].

In dieser Arbeit werden die Erfolgsfaktoren betrachtet, die zum Markt- bzw. technischen Erfolg eines Projektes beitragen. Der Markterfolg wird anhand Metriken gemessen wie: Downloads, Sterne auf GitHub sowie die Anzahl der Nutzer. Der technische Erfolg wird anhand von Metriken wie: Anzahl der Commits, Anzahl der Mitwirkenden am Projekt, Geschwindigkeit in der Tickets abgearbeitet werden und Verhältnis zwischen offenen/abgeschlossenen Issues gemessen.

2 Erfolgsfaktoren

Es gibt viele Faktoren, die Einfluss auf den Erfolg eines Projektes haben. In diesem Kapitel werden die Faktoren in zwei Klassen aufgeteilt und näher erklärt. Die *Haupterfolgsfaktoren* werden in dieser Bachelorarbeit mittels Datenerhebung und Umfrage analysiert. *Weitere Faktoren* werden zusätzlich auf Basis von Literatur betrachtet und diskutiert. Zu den *Haupterfolgsfaktoren* gehören Eigenschaften wie *Lizenzen*, *Qualität*, *Dokumentation*, *Community* und *Network Effekt* (ggf. noch anpassen). Diese Faktoren alleine schaffen allerdings kein Gesamtbild des Erfolgs, deshalb wird im Kapitel 2.2 auf zusätzlich Aspekte eingegangen. Diese werden allerdings nicht mittels Datenerhebung oder Umfrage analysiert, da der Aufwand diese empirisch zu erfassen zu hoch wäre, stattdessen findet die Analyse auf rein literarischer Ebene statt. Hierzu gehören Faktoren wie *Das richtige Timing*, *Modularität und Komplexität*, *Responsibility Assignment* (ggf. noch anpassen)

2.1 Haupterfolgsfaktoren

TODO: Einleitender Satz für Kapitel 2.1

2.1.1 Lizenzen

Laut Subramaniam et al. spielen Lizenzen eine signifikante Rolle für den Erfolg von Open Source Software. Freie Lizenzen wie MIT oder BSD haben einen positiven Einfluss vor allem auf Software Entwickler. Entwickler die OSS nutzen tun dies, um es in eigene Projekte einzubauen, gegebenenfalls zu modifizieren und weiterzuverbreiten. Das ist mit restriktiven Lizenzen wie GPL meist nicht bedingungslos umsetzbar. Restriktive Lizenzen wie GPL wirken sich daher negativ bis neutral auf den Erfolg von OSS aus. Wenn die Software allerdings an Endnutzer gerichtet ist, wie zum Beispiel die Chat-App Telegram, spielt die Lizenz eine weniger wichtige Rolle, da Weiterverbreitung und Modifizierung für diese Nutzergruppe keine Rolle spielen [Sub09].

Stewart et al. widerspricht der zweiten Aussage von Subramaniam et al. laut ihnen haben nicht-restriktive-Lizenzen nicht nur auf das Entwicklerinteresse, sondern auch auf das Nutzerinteresse einen positiven Einfluss. Während restriktive Lizenzen einen nicht signifikanten Einfluss auf Entwickleraktivität hätten. [Ste06]

Laut Midha und Palvia wirken sich freie Lizenzen positiv auf den Markterfolg aus, allerdings nur zu Beginn eines Projektes. Restriktive Lizenzen wiederum wirken sich negativ auf den technischen Erfolg aus [Mid12].

Ich glaube, dass sich offene Lizenzen durchgängig positiv auf ein Projekt auswirken. Offene Lizenzen werden tendenziell eher von Unternehmen verwendet als Projekte mit restriktiven Lizenzen, das führt zu einem dazu, dass die Beliebtheit und Bekanntheit des Projektes steigt, als auch die Wahrscheinlichkeit dass die Unternehmen zum Open Source Projekt etwas beitragen oder Sponsoren werden.

H 1. *Offene Lizenzen haben positiven Einfluss auf den Markterfolg.*

Steigt die Beliebtheit eines Projekts, so steigt auch das Interesse von Open Source Entwickler an einem renommierten Projekt zu arbeiten.

H 2. *Offene Lizenzen haben positiven Einfluss auf den technischen Erfolg.*

2.1.2 Gute Dokumentation

TODO: Blauen Text ausformulieren

TODO: Quelle für raussuchen sieh unten

TODO: Weitere Quelle einbauen [Dag10]

Dokumentationen spielen eine entscheidende Rolle für den Erfolg eines Projekts. Ohne eine gute Dokumentation ist die Software schwer zugänglich für die Benutzer und damit teils unbrauchbar. Mailing Listen und StackOverflow können eine gute Ergänzung zur Dokumentation sein, allerdings kann diese dadurch nicht ersetzt werden.

[Mögliche Quelle: Könnte aus [Ban13] stammen]

Mit einem Crawler ist es schwer zu beurteilen, ob eine Dokumentation gut ist oder nicht oder ob eine Dokumentation überhaupt existiert, da sich diese häufig auch auf der Homepage des Projekts befinden. Man kann aber Dokumentation mit als Punkt in die Umfrage mit aufnehmen.

GitHub's 2017 Open Source Survey showed incomplete or confusing documentation is the biggest problem for open source users. Good documentation invites people to interact with your project. Eventually, someone will open an issue or pull request. Use these interactions as opportunities to move them down the funnel.

Source: <https://opensource.guide/building-community/#make-people-feel-welcome>

Weitere hilfreiche Quellen: [GitHub Open Source Survey](#), [Blog Post](#) über Dokumentieren

"Wie wichtig ist eine gute Dokumentation bei der Auswahl einer OSS für Sie?" eignet sich als hervorragende Frage in der Umfrage. Datenerfassung aus GitHub muss ggf. händisch stattfinden und soll wie folgt kategorisiert werden:

- 0 = keine Dokumentation
- 1 = Basis Dokumentation in der GitHub README
- 2 = Ausführliche Dokumentation (Eigene Website / Code Beispiele etc.)

TODO: Hypothese schöner formulieren

H 3. *Dokumentation ist wichtig vorallem wenn die Zielgruppe Entwickler sind. (Gute Doku => Hoher Technischer Erfolg)*

2.1.3 Eine Community Aufbauen

TODO: Hier braucht es noch etwas Literatur Recherche.

There are two types of community, User Community and Developer Community...
Code of Conduct passt hier rein

2 Erfolgsfaktoren

Ein Open Source Projekt braucht eine Community. Eine Community von Benutzern und eine Community von Contributor. Ohne eine Community kann ein Projekt nicht wachsen. Contributor werden gebraucht um das Projekt kontinuierlich zu verbessern, Benutzer um es natürlich zu nutzen (aka Goal of the Success ") aber auch um Bugs zu finden und zu reporten, dies muss allerdings auch aktiv encouraged werden. [Ban13] Sprich die Entwickler müssen sich um die Community kümmern, bzw aktiv dafür sorgen, dass die Community wächst.

Todo

Weitere Quellen: [Mid12]

Mögliche weitere Quelle [How do Firms Make Use of Open Source Communities](#)

Wie in 2.1.3 später genauer erläutert wird, ist eine Community ein essenzieller Bestandteil für ein Open Source Projekt. Abhängig der Lizenz zieht man unterschiedliche Personengruppen an. Offene Lizenzen wie MIT lädt vor allem X an... Mit eingeschränkten Lizenzen wie GPL zieht man weniger Leute/Unternehmen etc. an und hindert somit das Wachstum der eigenen Community [Mögliche Quelle: [Ste06] PDF S. 16] Unternehmen nutzen die Software, eingie improven die die Software und ein Teil davon gibt zur OSS Community auch wieder zurück [Ban13]

2.1.4 Sponsoren

2.1.5 Schneeball Effekt / Network Effekt

TODO: Network Effekt recherchieren

[Mid12] in Kapitel 6.3 bzw die Hypothese H3a.

Direkt Zitat aus [Mid12] Kapitel 6.3.

As hypothesized in H3a, OSS projects that have a larger previous user base are more popular. This was true at all stages.

2.2 Weitere Faktoren

TODO: Einleitender Satz für Kapitel 2.1

2.2.1 Der richtige Zeitpunkt

Zitat aus Kapitel 3.1 hernehmen und Paraphrasieren/einbauen, siehe auch Schluss von 3.1

Direkt Zitat aus [Ban13] Kapitel 3.1

An interesting point made in Malcolm Gladwells book Outliers: The Story of Success [24] is that people are successful if their skills support products in a marketplace that is just maturing and where there is, consequently, still little competition. The same is certainly true for open source software projects as well: Projects that pick up a trend too late will have a difficult time thriving in a market that already supports other, large and mature projects

2.2.2 Modularität und Komplexität

Bezüglich Modularität gibt es hier etwas mehr [Mar15] siehe Kapitel IV. A) beziehungsweise die Quellen [42,43,49] in [Mar15]

2.2.3 Responsibility Assignment

2.2.4 More ?

Literaturverzeichnis

- [Ban13] W. Bangerth und T. Heister. What Makes Computational Open Source Software Libraries Successful? *Computational Science & Discovery*, 6(1):015010, Nov. 2013.
- [Dag10] B. Dagenais und M. P. Robillard. Creating and Evolving Developer Documentation: Understanding the Decisions of Open Source Contributors. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE '10, S. 127–136. Association for Computing Machinery, New York, NY, USA, Nov. 2010.
- [Mar15] D. Margan und S. Čandrlić. The success of open source software: A review. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, S. 1463–1468. 2015.
- [Mid12] V. Midha und P. Palvia. Factors Affecting the Success of Open Source Software. *Journal of Systems and Software*, 85(4):895–905, Apr. 2012.
- [Ope] The Open Source Definition | Open Source Initiative. <https://opensource.org/osd>.
- [Sta] Stack Overflow Developer Survey 2021. https://insights.stackoverflow.com/survey/2021/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2021. Zuletzt aufgerufen am 20.03.2022.
- [Ste06] K. J. Stewart, A. P. Ammeter und L. M. Maruping. Impacts of License Choice and Organizational Sponsorship on User Interest and Development Activity in Open Source Software Projects. *Information Systems Research*, 17(2):126–144, 2006.
- [Sub09] C. Subramaniam, R. Sen und M. L. Nelson. Determinants of Open Source Software Project Success: A Longitudinal Study. *Decision Support Systems*, 46(2):576–585, Jan. 2009.
- [W3T22] Usage Statistics and Market Share of Web Servers, February 2022. https://w3techs.com/technologies/overview/web_server, Febr. 2022. Zuletzt aufgerufen am 11.02.2022.