

# 1 Einleitung

**TODO:** Einschränkungen auf Packages. Grund: a) npm b) kleinerer Umfang für BA

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 zum Erfolg 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.

## 1.1 Ziel der Arbeit

**TODO:** Wird im Laufe der Arbeit genauer spezifizieren.

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.

## 1.2 Erfolg definieren

### Todo

Erfolg muss genauer definiert werden (Sieh Mail von Beneken am 16.02.). Sieh [Mid12] Kapitel 2.1, sowie die zitierten Quellen in diesem Kapitel.

Um Erfolg in Open Source Software (OSS) messen zu können muss Erfolg zunächst definiert werden. Nach Midha lässt sich Erfolg in zwei Klassen unterteilen, den Markterfolg und den technischen Erfolg. Für ein vollständiges Gesamtbild des Erfolges müssen und werden beide Klassen miteinbezogen [Mid12].

### 1.2.1 Markterfolg

Markterfolg wird durch Charakteristika wie Beliebtheit gekennzeichnet. Diese Eigenschaft spiegelt sich beispielsweise in der Anzahl der Nutzer, GitHub Sterne oder Downloads wider.

## *1 Einleitung*

Einige dieser Metriken finden sich auf den GitHub Seiten der jeweiligen Projekte wieder und werden vom Webcrawler erfasst. [Mid12].

### **1.2.2 Technischer Erfolg**

Unter dem technischen Erfolg zählen Eigenschaften wie Frequenz von Updates, sowie Anzahl an Commits und Mitwirkenden. Diese Daten lassen sich ebenfalls auf GitHub erfassen. [Mid12].

## 2 Erfolgsfaktoren

---

### Wie soll man jetzt in diesem Kapitel die Faktoren unterteilen?

Aus den ganzen Erfolgsfaktoren sollen folgende Punkte spezifisch bearbeitet werden. Weitere Punkte können zusätzlich auf Basis von Literatur angesprochen werden. Sieh Liste weiter unten

1. Lizenzen haben einen Einfluss auf den Erfolg der Projekte
2. Hohe Qualität der Software => positiver Einfluss aufs Projekt
3. Gute Dokumentation => positiver Einfluss aufs Projekt
4. Wer sich um die Community kümmert (devs / userbase) => positiver Einfluss aufs Projekt

Diese Punkte können angesprochen werden, sind aber etwas schwer zu erfassen mit einer Umfrage oder Crawler.

- Network Effekt, an der Community kann man zwar arbeiten den Network Effekt aber nicht so richtig triggern (I guess)
- Responsibility Assignment (Bezug auf Community / Qualität)
- Timing
- Modularität
- Komplexität

**BTW** Wenn es aussieht als ob die Arbeit zu kurz wird, kann man noch einzelne Projekte ganz genau unter die Lupe nehmen I guess z.B. Linux, oder React for that matter weil JavaScript etc.

React gegenüberstellen zu Vue. Das eine ist ja aus dem Hause Facebook, das andere völlig Open Source... Weitere mögliche Kapitel also *Case Study: React vs. Vue* oder so ähnlich

---

**TODO:** Infos ergänzen nachdem, die Unterkapitel geschrieben sind

Im Artikel von Wolfgang Bangerth und Timo Heister werden die Erfolgsfaktoren in zwei Klassen unterteilt. Eine Klasse mit beeinflussbaren Faktoren und eine für nicht beeinflussbare Faktoren [Ban13]. Im Folgenden sollen diese Faktoren erklärt werden und wie sie in diese Arbeit eingebunden werden.

### 2.1 Beeinflussbare Faktoren

Beeinflussbare Faktoren sind die, die OSS Entwickler selbst in der Hand haben. Zu diesen Faktoren gehört die Lizenz, Qualität der Software sowie der Dokumentation und der [Aufbau](#) einer Community.

#### 2.1.1 Lizenzen

*(Der folgende Teil ist nur eine Ideensammlung und noch nicht ausformuliert.)*

Laut Midha und Palvia spielt die Lizenz eine **insignifikante** Rolle beim Erfolg eines Projektes. Lediglich zu Beginn eines Projektes spielt die Lizenz eine Rolle, da für den Nutzer wenige alternative Metriken zur Entscheidung stehen. Die Begründung hierfür sei das, wenn ein Projekt schon beliebt ist die Beliebtheit höher gewichtet wird als die Lizenz. Gleichzeitig heißt es aber auch, dass GPL und ähnlich restriktive Lizenzen einen positiven Effekt auf die Entwickler haben. Vor allem im späteren Verlauf der Entwicklung [Mid12]

[Sub09] widerspricht zum Teil [Mid12], laut Subramaniam et. al. haben Lizenzen einen signifikanten Einfluss auf den Erfolg von Open Source vor allem dann, wenn die Zielgruppe Entwickler sind. Restriktive Lizenzen haben einen negativen Effekt auf den Erfolg von OSS. Die Erklärung von Subramaniam [ist/sei](#), dass Entwickler OSS nutzen mit der Intention es zu modifizieren, in eigene Projekte einzubauen und weiterzuverbreiten, dies ist mit restriktiven Lizenzen meist nicht bedingungslos umsetzbar. Wenn die Software allerdings an Endnutzer gerichtet ist, spielt die Lizenz eine insignifikante Rolle, da Weiterverbreitung und Modifizierung für den normalen Nutzer keine Rolle spielen [Sub09].

**TODO: Kommenden Absatz schöner Formulieren**

Wie in 2.1.4 später genauer erläutert wird, ist eine Community wichtig, abhängig der Lizenz kann man sich hier entweder das Leben schwer machen oder nicht, eine offene Lizenz wie MIT lädt alle dazu ein beizutragen. Unternehmen nutzen die Software, einige verbessern die Software und ein Teil davon gibt zur OSS Community auch wieder zurück [Ban13] Mit eingeschränkten Lizenzen wie GPL zieht man weniger Leute/Unternehmen etc. an und hindert somit das Wachstum der eigenen Community

[Der Wechsel von Docker könnte ggf. Interessant sein sich anzuschauen. Hier \[Del03\] hab ich was gelesen und markiert, aber bisher als Quelle noch nicht eingebaut](#)

**H 1.** *Lizenzen haben einen signifikanten Einfluss auf den technischen Erfolg. Wobei offene Lizenzen wie MIT o.ä. einen positiven, während restriktive Lizenzen einen eher negativen Einfluss haben.*

Die Datenerhebung wird mittels Datenanalyse und Teil der Umfrage sein.

#### 2.1.2 Qualität

*(Der folgende Teil ist nur eine Ideensammlung und noch nicht ausformuliert.)*

Ein weiterer Punkt worüber Entwickler Einfluss haben ist die Qualität der Software. Produkts als auch die User Experience (UX) beziehungsweise Developer Experience (DX). Sprich, gibt es eine gute Dokumentation, ist das Package etc. Customizable nach den Wünschen des Nutzers etc. Zu der Qualität gehört auch Software die "Bug-Frei"läuft, oder zumindest die Funktionalität die Bugs überwiegt [Ban13].

Direkt Zitat aus [Ban13] Kapitel 2.1

[W]ithout a focus on fixing bugs as soon as they are identified will never be of high quality. Thus, quality needs to be an important aspect of development from the start.

Die *First Time Experience* spielt eine wichtige Rolle, ist ein Tool schwer aufzusetzen / installieren (bei npm trifft das nicht ganz zu?) beziehungsweise die Dokumentation nicht schlüssig genug, gibt es meist eine (Hand voll/Menge...) alternative Tools die ein User stattdessen einfach hernehmen kann, statt sich mit Tool X herumzuschlagen [Ban13].

Als Beispiel, könnte man hierfür sowas nehmen wie eine JS Bibliothek die 10 verschiedene Alternativen hat. Beispielsweise hat das Tool X mit 300k Downloads 5 alternativen, die zwar nicht so viele Download zahlen haben, aber wenn man das Tool nicht zum laufen kriegt, dann schlägt man sich damit auch nicht weiter rum und geht einfach zu Y, Z oder A.

**H 2.** Die Qualität spielt eine sehr wichtige Rolle, vor allem dann, wenn es Alternativen gibt. (Gute Qualität => Guter Technischer/Markt Erfolg)

### 2.1.3 Gute Dokumentation

*(Der folgende Teil ist nur eine Ideensammlung und noch nicht ausformuliert.)*

Dokumentationen spielen eine entscheidende Rolle beim Erfolg eines Projekts. Ohne eine gute Dokumentation ist die Software schwerer zugänglich für die Benutzer und damit teils unbrauchbar, ausgenommen Projekte mit intuitiven User Interfaces. Mailing Listen und StackOverflow können eine gute Ergänzung zur Dokumentation sein, allerdings kann diese dadurch nicht ersetzt werden. *Eine gute Dokumentation skaliert auch besser, egal wie viele neue User das Projekt hat, alle können mit der Doku anfangen und das Produkt nutzen, andernfalls würde man vor lauter Fragen beantworten nicht mehr hinterherkommen. (Eher ne Idee Satz kann auch komplett entfernt werden.)* 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. [Mögliche Quelle: Könnte aus [Ban13] stammen]

"Wie wichtig ist eine gute Dokumentation bei der Auswahl einer OSS für Sie?" eignet sich als hervorragende Frage in der Umfrage.

Alternative könnte man diese Daten auch erfassen, allerdings nur von Hand. Da zum einen die Dokumentationen nicht immer in der README.md sind, sondern auf anderen Website und der Crawler nicht beurteilen kann, ob eine Dokumentation gut ist oder nicht. Daher könnte man quasi eine Liste zum Abhacken durchgehen Beispielsweise wie folgt:

- Hat das Projekt eine Dokumentation?
- Hat die Doku Anwendungsbeispiele?
- Ist ein Sandbox-Modus für dieses Projekt möglich? Wenn ja, gibt es einen?

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

### 2.1.4 Eine Community Aufbauen

Braucht es dieses Kapitel zwingend? Sieh Kapitel 2.3. in [Ban13]

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 reporte, dies muss allerdings auch aktive encouraged werden. [Ban13]

**H 4.** *Bemühungen darin eine Community aufzubauen führen zu einem hohen Markt Erfolg.*

## 2.2 Nicht beeinflussbare Faktoren

Nicht beeinflussbare Faktoren sind Faktoren über die OSS Entwickler wenig bis gar kein Einfluss haben.

### 2.2.1 Schneeball Effekt / Network Effekt

**TODO:** Network Effekt recherchieren

Eine zweite Quelle bestätigt das Vox Populi. In [Sub09] wird vom sogenannten *Network Effekt* gesprochen. Dieser wirkt sich laut Subraminam et. al. positiv auf den Erfolg von OSS aus. [Network Effekt in 2. Worten erklären... den Aktiven als auch den Passiven](#)

Direkt Zitat aus [Sub09] Kapitel 5.1.

The results from our study show the important role played by network effects of OSS

**H 5.** *Erfolgreiche Projekte werden noch erfolgreicher. => neue Projekte müssen raus stechen sonst haben sie keine Chance gegen die bestehenden OSS Alternativen*

### 2.2.2 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

## 3 Crawler

Mithilfe des Crawlers kann sowohl der Markterfolg als auch der technische Erfolg gemessen werden. Über Metriken wie GitHub Sterne, Forks oder Downloads lässt sich der Markterfolg quantifizieren. Der technische Erfolg findet sich wieder in Daten wie Anzahl an Commits, Frequenz von Commits oder Mitwirkende.

Bei der ersten Recherche wurde festgestellt, dass die Repositories mit den meisten Github Sternen keine Software direkt sind, sondern Repositories mit Lerninhalten. Zum Zeitpunkt des Schreibens (13. Februar 2022) werden Platz eins, drei und vier von

- freeCodeCamp (340k Sterne),
- free-programming-books (222k Sterne) und
- coding-interview-university (209k Sterne)

belegt.

Insgesamt sind unter den Top 10 Repositories auf GitHub nur drei IT Projekte im eigentlichen Sinne dabei [Top].

- Vue (193k Sterne),
- React (182k Sterne) und
- TensorFlow (163k Sterne).

Es wird daher nicht möglich sein die Top 100 Projekte für die Datensammlung zu verwenden. Stattdessen werden Projekte von Hand ausgewählt, um zu garantieren, dass es sich nur um IT Projekte handelt.

### 3.0.1 Der Crawler

Hilfreiche Ressourcen:

- Welche Daten könnte man erfassen und wie? Sieh [npms.io](https://npms.io).
- Hilfreiches Tool: [npmrends.com](https://npmrends.com)
- API um an NPM Downloads ran zu kommen: [GitHub registry](#) bzw. [Deprecated registry](#)

### 3.0.2 Disclaimer

npm downloadzahlen sind manipulierbar, ich geh zwar davon aus, dass es allgemein nicht häufig gemacht wird und da ja sowieso vielmehr in die rechnung rein geht als nur npm downloads. Werden diese daten trotzdem mit erfasst. [How to exploit NPM](#).

In einem [Blogpost](#) erklärt NPM wie es zu den Download zahlen zustande kommt.

# Literaturverzeichnis

- [Ban13] W. Bangerth und T. Heister. What Makes Computational Open Source Software Libraries Successful? *Computational Science & Discovery*, 6(1):015010, Nov. 2013.
- [Del03] W. H. Delone und Ephraim R. McLean. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*, 19(4):9–30, 2003.
- [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\\_share&utm\\_medium=social&utm\\_campaign=dev-survey-2021](https://insights.stackoverflow.com/survey/2021/?utm_share&utm_medium=social&utm_campaign=dev-survey-2021).
- [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.
- [Top] Top GitHub Projects by Stars. <https://github.com/search?l=&o=desc&q=stars%3A%3E100+stars%3A%3E1&s=stars&type=Repositories>.
- [W3T22] Usage Statistics and Market Share of Web Servers, February 2022. [https://w3techs.com/technologies/overview/web\\_server](https://w3techs.com/technologies/overview/web_server), Febr. 2022. Zuletzt aufgerufen am 11.02.2022.