

# Standard for Public Code

Handreichung für die Entwicklung und Nachnutzung von Code im öffentlichen Interesse

Vorlage für das ISO, ITU, BSI und DIN



Entwurf 0.99

Stand: 29.5.2020

Olaf-Gerd Gemein, oggemein@googlemail.com

Lizenz: cc0 1.0 Universal -><https://creativecommons.org/publicdomain/zero/1.0/>

Contributions Oliver Rack, Verus von Plotho, Peter Muryskin

## Inhaltsverzeichnis

<b>Einführung</b>	<b>3</b>
<b>Einhaltung von Standards oder Zertifizierungsverfahren</b>	<b>9</b>
<b>Die Ziele des Standard für Public Code</b>	<b>11</b>
<b>Leitfaden für Leser</b>	<b>13</b>
<b>Glossar</b>	<b>15</b>
<b>1: Offener Code</b>	<b>17</b>
<b>2: Bundle Policy und Quellcode</b>	<b>19</b>
<b>3: Erstellen Sie wiederverwendbaren und portablen Code</b>	<b>21</b>
<b>4: Beiträge willkommen heißen</b>	<b>23</b>
<b>5: Versionskontrolle pflegen</b>	<b>25</b>
<b>6: Überprüfung der Beiträge verlangen</b>	<b>27</b>
<b>7: Dokumentieren Sie Ihre Ziele</b>	<b>29</b>
<b>8: Verwenden Sie einfaches Englisch</b>	<b>31</b>
<b>9: Offene Standards verwenden</b>	<b>33</b>
<b>10: Kontinuierliche Integration betreiben</b>	<b>35</b>
<b>11: Unter einer offenen Lizenz veröffentlichen</b>	<b>37</b>
<b>12: Einen zusammenhängenden Stil verwenden</b>	<b>39</b>
<b>13: Achten Sie auf die Reifegrad der Codebase</b>	<b>41</b>

## Einführung

Der Standard for Public Code ist eine Auswahl von Kriterien, die öffentliche Einrichtungen bei der gemeinsamen Entwicklung und Pflege von Software und Richtlinien unterstützt.

Jeder, der Software oder Richtlinien für einen öffentlichen Zweck entwickelt, kann diesen Standard nutzen, um auf qualitativ hochwertige öffentliche Dienste hinzuarbeiten, die kosteneffizienter sind und weniger Risiko sowie mehr Kontrolle bieten. In 2020 wird auf dieser Basis eine DIN SPEC (PAS) entwickelt, zu welcher zu einem öffentlichen Diskurs eingeladen werden wird. Auch auf internationaler Ebene gibt dazu bereits Bestrebungen, die als Ergänzung im gleichen Zusammenhang gesehen werden können, wie z.B. das OpenChain Projekt der Linux Foundation, oder

Diese Einführung beschreibt den Begriff öffentlicher Code, erklärt, warum dies wichtig ist, und stellt einen möglichen Prozess vor, durch den Software- und Richtliniencode zu zertifizierten öffentlichen Code werden kann. Dies erfolgt zum Einen vor dem Hintergrund, auch in Deutschland eine neutrale und allgemein anerkannte Zertifizierungsstelle zu etablieren, als auch als Diskussionsbeitrag zur öffentlichen Diskussion.

## Definition von öffentlichem Code

Öffentlicher Code ist sowohl Zivil-Code (wie Gesetze, Richtlinien oder Vorschriften) als auch Computer-Quellcode (wie Software und Algorithmen), der in einem öffentlichen Kontext, von Menschen oder Maschinen, ausgeführt wird. Öffentlicher Code unterscheidet sich ausdrücklich von normaler Software, weil er unter grundlegend anderen Umständen und Erwartungen funktioniert und eng mit der staatliche-gesellschaftlichen Verantwortung und Daseinsvorsorge verknüpft ist.

Öffentlich relevant sein könnte auch "Produktionscode", also Bauanweisungen für die Herstellung von Gütern im öffentlichen Raum (z.B. für Verkehrssteuerung), sowohl für Hardware als auch für Software. Darauf werden wir in einer erweiterten Version eingehen. Auch die aktuelle Pandemie zeigt auf, was wir in diesem Zusammenhang zukünftig bedenken sollten, wie zum Beispiel die irische Open Hardware Initiative für die Beatmungsgeräte -> "Open Source Ventilator"<sup>1</sup>.

## Warum öffentlicher Code?

Es gibt viele gute Gründe dafür, warum öffentlicher Code aktuell besonders relevant ist.

### **Software-Code == gesetzliche Anweisungen und Verfahren digitalisiert**

Software ist somit ein Teil der öffentlichen Infrastruktur.

Im 21. Jahrhundert kann Software als wichtige öffentliche Infrastruktur betrachtet werden. Sie ist zunehmend nicht mehr nur Ausdruck der bestehenden Politik, sondern auch der Urheber einer neuen Politik - zum Beispiel wenn Algorithmen entscheiden, welche Bezirke

---

<sup>1</sup> <https://opensourceventilator.ie>

zusätzliche soziale Dienste oder Polizeiarbeit benötigen. Dazu besteht auch in Deutschland eine breite und intensive Diskussion.

Software-Frameworks, Algorithmen und Datenerfassung sind zu Schlüsselementen bei der Ausführung öffentlicher Richtlinien geworden. Computercode führt nun Richtlinien aus, die im gesetzlichen Code durch demokratische Verfahren kodifiziert wurden. Beide Formen von Code stellen Bedingungen für das Funktionieren der Gesellschaft nach demokratisch festgelegten öffentlichen Werten auf, wobei letztere von Menschen, erstere von Maschinen ausgeführt werden. Mit anderen Worten: Software-Code hat zunehmend damit begonnen, Rechtscode zu ersetzen bzw. zu substituieren. Die Umsetzung des OZG bis 2022 wird alle relevanten Fachverfahren betreffen, und die Digitalisierung der Prozesse wird die heutige Auffassung und den Status Quo manifestieren und in "Software gießen". Diese Algorithmen werden das Recht umsetzen, wie es aktuell kodifiziert ist, und die weiteren Verfahren werden mit Hilfe von unterstützenden "Maschine-learning" und künstlicher Intelligenz auf diese kodifizierten Strukturen aufsetzen.

Software sollte daher den Prinzipien der demokratischen Regierungsführung unterworfen werden, und gehört allenthalben zum Bereich der "öffentlichen Daseinsvorsorge".

### Traditionelle öffentliche Softwarebeschaffung

Die bisherigen Methoden der öffentlichen Softwarebeschaffung erscheinen unter diesen Aspekten wenig hilfreich und haben bei der Erbringung öffentlicher Dienstleistungen in der Vergangenheit häufig keinen oder geringen Nutzen gebracht - die Gründe dafür sind ebenso vielfältig wie die Folgen tiefgreifend. Heute existiert eine nicht mehr zu kontrollierende, inkonsistente und sehr eingeschränkt verwertbare Vielfalt an Daten, Software, Betriebssystemen.

In den letzten zehn Jahren waren öffentliche Einrichtungen, die komplette Softwarelösungen gekauft haben, manchmal überrascht, zu entdecken, daß sie:

- Software nicht beeinflussen, also "umprogrammieren" können, um sich ändernden Richtlinien zu folgen oder Vorteile neuer Technologien zu nutzen.
- keinen Zugang zu ihren Daten haben, da diese in proprietären Systemen gespeichert sind
- aufgefordert werden, immer höhere und immer wieder Lizenzgebühren zu zahlen

### Technologische Souveränität und demokratische Rechenschaftspflicht

Unsere Gesellschaft verdient Besseres, es besteht Handlungsbedarf. Nicht nur das OZG, auch die aktuellen Herausforderungen im Rahmen der Digitalisierung bedürfen ein "Upgrade" in der Herangehensweise und Umgang mit "öffentlicher Software".

Wir glauben, dass Public-Software, die unsere Gesellschaft zunehmend beeinflusst, nicht länger eine Black Box sein kann, die von externen Unternehmen kontrolliert wird, die die zugrunde liegende Logik, auf der ihre Software arbeitet, in proprietären Codebasen versteckt halten, und damit praktisch einer transparenten und öffentlichen Kontrolle entzieht oder diese zumindest massive erschwert.

Stattdessen brauchen öffentliche Verwaltungen, Regierungen sowie die Menschen, denen sie dienen, technologische Souveränität - die es ihnen erlaubt, das Funktionieren öffentlicher Software partizipativ mitzugestalten, die Verfahren transparent festzulegen und zu kontrollieren, vergleichbar so wie sie in der Lage sind, eine Richtlinie festzulegen und zu kontrollieren, die gesetzlich formuliert ist. Bürger und zivilgesellschaftliche Akteure brauchen eine neue Softwarequalität (und neue Verfahren zu deren Beschaffung und Erstellung), um transparent und rechenschaftspflichtig zu sein. Das Design von Software als wesentliche Infrastruktur sollte neu geregelt. Die nachfolgenden Richtlinien sind das Ergebnis eines öffentlichen Dialogs in den letzten Jahren in den Niederlanden<sup>2</sup> mit der Foundation für Public Code und mit der Open und Agile Smart Cities Initiative<sup>3</sup> aus Brüssel. Weite Teile dieser Handreichung sind in der gemeinsamen Arbeit entstanden und daraus entnommen<sup>4</sup>. In Deutschland sind die Städte Heidelberg, Kiel, Paderborn, Delbrück und Wolfsburg Teil der internationalen Städtegemeinschaft.

## Wirklich öffentliche Software entwerfen

Der öffentliche Code ist das Kernstück moderner öffentlicher Einrichtungen, prägt die Arbeit der Angestellten und Beamten und beeinflusst zunehmend das Leben fast aller Einwohner.

Öffentliche Software muss daher sein:

- transparent
- verantwortlich
- verständlich für Bürger
- technologisch souverän gespeichert und bereitgestellt

Sie muss auch jeweils aktuell die Werte der Gesellschaft widerspiegeln, der sie dient, indem sie beispielsweise integrativ und nicht diskriminierend ist sowie durch kollaboratives Einreichen von kodifizierbaren Änderungsvorschlägen und Fehlerkorrekturen eine öffentliche Diskussion und Zusammenarbeit ALLER Stakeholder ermöglicht und fördert .

Die meisten proprietären Softwaresysteme, die derzeit von öffentlichen Einrichtungen verwendet werden, erfüllen diese Anforderungen nur sehr unzureichend. Öffentlicher Code (Software), die als öffentliche Infrastruktur anzusehen ist, muss von Beginn an ...

Auch das OZG bietet heuer in Deutschland dazu eine einmalige Gelegenheit, versäumtes nachzuholen und sich für die Zukunft aufzustellen.

---

2

3

4

## Die Werte des öffentlichen Codes

Wir sind der Ansicht, dass öffentlicher Code folgende Merkmale haben muss, wenn er die Grundwerte unserer demokratischen Gesellschaft in sich tragen soll, weil er der Daseinsvorsorge gewidmet ist:

- Inklusiv
- Nutzbar
- Offen
- Maschinenverstehbar und Lesbar
- Durch Akzeptanz-Tests in seiner Funktion und Qualität nachvollziehbar
- Verantwortlich
- Zugänglich
- Nachhaltig

## Wie öffentlicher Code funktioniert

Öffentlicher Code ist deshalb wenn irgend möglich als Open-Source-Software zu gestalten, die für die Erfüllung der wesentlichen Rolle von öffentlichen Einrichtungen gedacht ist. Durch die Erstnutzung tragen andere Verwaltungen und sonstige Akteure wie z.B. aus Wissenschaft u. dem "digitalen Ehrenamt" wieder zur Weiterentwicklung der Codebasis bei, so dass ihre Entwicklung und Wartung wirklich kollaborativ werden und nachnutzbar (als portierbar, replizierbar und interoperabel) werden.

### **Offen zu sein macht Neues möglich!**

Lokale Verantwortung und demokratische Rechenschaftspflicht sind gewährleistet, wenn eine öffentliche Organisation (insbesondere auch einschließlich unmittelbarer Behörden und Unternehmen wie z.B. Anstalten des öffentlichen Rechtes) ihren eigenen öffentlichen Code auf der Grundlage der in den unter beschriebenen Richtlinien niedergelegten Grundsätzen erstellt oder erstellen lässt, ebenso offen implementiert und pflegt.

Durch Offenheit und eine breitere Basis von Mitwirkenden ist die Software sicherer - sie profitiert davon, dass viele Augen mögliche Fehler erkennen. Viele Mitwirkende teilen sich die Wartungsarbeit, um sie funktionsfähig und modern zu halten, was die zukünftige technische Fehler verringert. Beispiele für eine solche "Implementierung Partnerschaft" gibt es inzwischen auch in Deutschland, z.B. beim Masterportal<sup>5</sup> oder den digitalen Modellregionen in NRW<sup>6</sup>

*„Vertrauenswürdige Cloud-Services lassen sich nur mit quelloffener Software realisieren. Langfristige Sicherheit und Herstellerunabhängigkeit für die Anwender bietet nur der Einsatz von quelloffener Software, da hier der menschenlesbare Quellcode eines Software-Programms öffentlich zugänglich ist. Denn nur wenn unabhängige Experten den Quellcode (Source Code) einer Software einsehen und überprüfen können, kann ausgeschlossen werden, dass vom Anbieter der Software*

---

5

6

*Hintertüren(Backdoors) eingebaut oder Malware in die Software eingeschleust wurde. Nur quelloffene Software liefert die notwendige Transparenz, um „blindes Vertrauen“ zu ersetzen“,*

sagte Rafael Laguna, Direktor der Bundesagentur für Sprunginnovationen am 14.2.2020 im Interview mit dem Tagesspiegel Berlin.

Die geteilte Arbeitsbelastung ist jetzt und in Zukunft nachhaltiger. Die Offenheit macht den Code und seine Daten in Zukunft leichter anpassbar - es wird einfacher sein, sie umzurüsten, neu zu nutzen oder auslaufen zu lassen. Dies alles führt zu einer risikoärmeren und selbstbestimmteren öffentlichen Infrastruktur und ist auch hinsichtlich der Ressourcensparsamkeit geboten.

Diese Bündelung von Ressourcen ermöglicht es den öffentlichen Verwaltungen, auch selbst der Anpassung der Software besondere Aufmerksamkeit widmen zu können, so dass sie in jedem lokalen Kontext am besten funktioniert - und so letztlich auch eine bessere Benutzererfahrung für ihre Endbenutzer (Einwohner oder Bürger) zu ermöglichen.

### Ökonomie des öffentlichen Codes

Öffentlicher Code bietet ein besseres Wirtschaftsmodell sowohl für öffentliche Einrichtungen als auch für kommerzielle Unternehmen. Es ist eine Alternative zur traditionellen Softwarebeschaffung, die die lokale Kontrolle und die wirtschaftlichen Möglichkeiten aller Beteiligten erhöht.

Öffentlicher Code sollte von Anfang an offen, anpassungsfähig und mit Datenportabilität konzipiert werden und kann sowohl von internen Mitarbeitern oder vertrauenswürdigen Anbietern erzeugt werden. Da der Code offen ist, kann die öffentliche Verwaltung bei Bedarf den Anbieter wechseln. Offener Code erhöht die Möglichkeiten für öffentliches Lernen und Kontrolle und ermöglicht es der öffentlichen Verwaltung, kleinere Aufträge zu beschaffen - und erleichtert so die Angebotsabgabe für lokale kleine und mittlere Unternehmen. Öffentliche Verwaltungen können ihre eigene Softwarebeschaffung nutzen, um Innovation und Wettbewerb in ihrer lokalen Wirtschaft zu fördern.

Dies kann als Investition angesehen werden, die zu zukünftigem Wirtschaftswachstum führt (Digitaldividende und Digitalallmende), aufgrund der wachsenden Nachfrage nach offener Technologie werden mehr Implementierungspartner notwendig sein. Eine strategisch gewollte Nachnutzung in der Bundesrepublik Deutschland ist ausdrücklich möglich und sollte durch entsprechende Angebote, wie Workshops, Webinare und ausführliche Dokumentationen begleitet werden.

### Beschaffung von öffentlichem Code

Öffentlicher Code kann von internen Entwicklungsteams, Auftragnehmern oder externen Lieferanten verwendet und entwickelt werden. Anbieter von öffentlichen Einrichtungen können öffentlichen Code in ihre Vertragsangebote aufnehmen.



Wenn Sie vorhandenen öffentlichen Code verwenden möchten, müssen Sie in Ihrem Budget und in Ihrem Projektentwurf angeben, dass Ihre neue Lösung diese Codebase verwendet. Um einen innovativen Ansatz zur Anpassung des öffentlichen Codes an Ihren Bedarf zu fördern, könnten Sie die Dienstleistung oder das Ergebnis in Ihrem Vertrag spezifizieren.

Bei der Ausschreibung im Rahmen der digitalen Modellregion OWL mit der Leit Kommune Paderborn ist explizit darauf geachtet worden, in der Ausschreibung auf nationale und internationale Standards zu verweisen sowie verfügbare Softwaremodule, welche also Open-Source-Software verfügbar sind, wiederzuverwenden.

Auch z.B. in Kiel hat man 2019 entsprechende Richtlinien<sup>7</sup> erlassen:

*“Die Ratsversammlung beschließt, dass künftig für alle neuen digitalen Lösungen der Stadt, welche mit smarter Sensorik, Software und/oder Daten arbeiten, folgende Leitlinien zu beachten sind:*

- *Der Einsatz intelligenter Sensorik sowie die dahinter stehende Soft- und Hardware wird mit den Bürger\*innen vor Ort, Maker-Spaces, Handwerk und anderen relevanten Akteuren gemeinsam geplant und entwickelt. Hierbei muss die generelle Stadtentwicklung mit gedacht und unterstützt werden.*
- *Die eingesetzte Hard- und Software soll unter einer freien Lizenz stehen (FLOSS). Alle dazugehörigen Konzepte und Projektdokumente werden von der Stadt unter freier Lizenz veröffentlicht. Abweichungen davon müssen gegenüber der Ratsversammlung begründet werden.*
- *Für neue Projekte und Lösungen soll immer zuerst bereits bestehende freie Software genutzt und weiterentwickelt werden. Im Vergabeverfahren soll dies als Vergabekriterium berücksichtigt werden. Eigene Entwicklungen oder der Kauf von closed-source-Software erfolgt nur, wenn es keine vergleichbaren freien Lösungen gibt.*
- *Erfolgreiche Smart City-Konzepte werden gemeinsam mit anderen Städten und Regionen in Europa und der Welt entwickelt und geteilt.*
- *Grundsätzlich werden alle generierten Daten der Bevölkerung datenschutzkonform aufbereitet zur Verfügung gestellt (Open Data).*
- *Der Einsatz intelligenter Systeme muss den Grundsätzen des europäischen Datenschutzrechts entsprechen – Privacy by Design wird von Anfang an in einer verpflichtenden Datenschutzfolgenabschätzung mitgedacht.*
- *Standards der IT-Sicherheit nach dem Bundesamt für Informationssicherheit müssen erfüllt werden.*
- *Um kleine und mittelständische Unternehmen aus Kiel in der Entwicklung und dem Kompetenzgewinn im Bereich von freier Software zu unterstützen, soll die Stadt Kiel diese, soweit möglich, in Projekte und Vorhaben einbeziehen.*

---

<sup>7</sup> <https://ratsinfo.kiel.de/bi/vo021.asp?VOLFDNR=23151>

Inzwischen hat auch München und Hamburg<sup>8</sup> nachgezogen, und in 2020 erwarten wir eine bundesweite neue Welle zugunsten Open Source im Sektor der Daseinsvorsorge und Smart Cities.

In 2019 hat auch die USA einen weiteren entscheidenden Schritt getan, und das “Federal Source Code Policy: M-16-21 MEMORANDUM FOR THE HEADS OF DEPARTMENTS AND AGENCIES”<sup>9</sup> erlassen. Dort heisst u.a.:

*“... können zusätzliche Vorteile entstehen, wenn der Quellcode auch der Öffentlichkeit als OSS zur Verfügung gestellt wird. Die Bereitstellung von Quellcode als OSS kann eine kontinuierliche Verbesserung von kundenspezifisch entwickelten Code-Projekten des Bundes ermöglichen, wenn eine breitere Nutzergemeinschaft den Code für ihre eigenen Zwecke implementiert und Verbesserungen veröffentlicht. Diese kooperative Atmosphäre kann die Durchführung von Software-Peer-Reviews und Sicherheitstests, die Wiederverwendung vorhandener Lösungen und den Austausch von technischem Wissen erleichtern<sup>10</sup>. Darüber hinaus können Anbieter, die an einer künftigen Wartung oder Verbesserung teilnehmen oder um diese konkurrieren, dies mit voller Kenntnis des zugrunde liegenden Quellcodes tun. Eine Reihe von Unternehmen des privaten Sektors haben bereits einige ihrer Software-Entwicklungsprojekte auf ein OSS-Modell umgestellt, bei dem der Quellcode der Software der Öffentlichkeit auf breiter Basis zur Einsicht, Verbesserung und Wiederverwendung zur Verfügung gestellt wird.”*

Bereits 2015 hat die USA unter der Obama Administration die OKF eingeladen an dem Projekt 18F<sup>11</sup> mitzuwirken, welches immer noch wegweisend ist hinsichtlich der praktischen Implementierung.

Im Februar 2019 hat die schweizerische Eidgenossenschaft den “Strategischen Leitfaden Open Source Software in der Bundesverwaltung”<sup>12</sup> veröffentlicht, und dabei ebenfalls klare Richtlinien beschrieben, welche die OSS in der Bundesverwaltung fördern und “Waffengleichheit<sup>13</sup>” mit proprietären Lösungen herstellen.

---

<sup>8</sup> <https://www.heise.de/news/Rot-Gruen-in-Hamburg-plant-Einstieg-in-den-Ausstieg-von-Microsoft-4770666.html>

<sup>9</sup> <https://sourcecode.cio.gov>

<sup>10</sup> Department of Defense Chief Information Officer. *Clarifying Guidance Regarding Open Source Software (OSS)*. October 16, 2009. “The continuous and broad peer-review enabled by publicly available source code supports software reliability and security efforts through the identification and elimination of defects that might otherwise go unrecognized by a more limited core development team.” <http://dodcio.defense.gov/Portals/0/Documents/FOSS/2009OSS.pdf>.

<sup>11</sup> 18F (<https://18f.gsa.gov/>) is a digital services delivery team within the General Services Administration. The 18F Open Source Policy is described at <https://18f.gsa.gov/2014/07/29/18f-an-open-source-team/> and can be accessed at <https://github.com/18F/open-source-policy/blob/master/policy.md>.

<sup>12</sup>

<https://www.isb.admin.ch/isb/de/home/ikt-vorgaben/strategien-teilstrategien/sb004-ikt-teilstrategie-open-source.html>

<sup>13</sup> “Waffengleichheit” in der Schweiz heisst dort “Gleiche Spießlänge”

## Einhaltung von Standards oder Zertifizierungsverfahren

In Europa stellt z.B. die Foundation für Public Code, Amsterdam, sicher, dass von ihr verwaltete Codebases mit dem Standard for Public Code wie er hier im Weiteren beschrieben ist, übereinstimmen. Dies macht potentiellen Benutzern und Mitwirkenden klar, dass die Codebase wie auch Aktualisierungen von hoher Qualität sind.

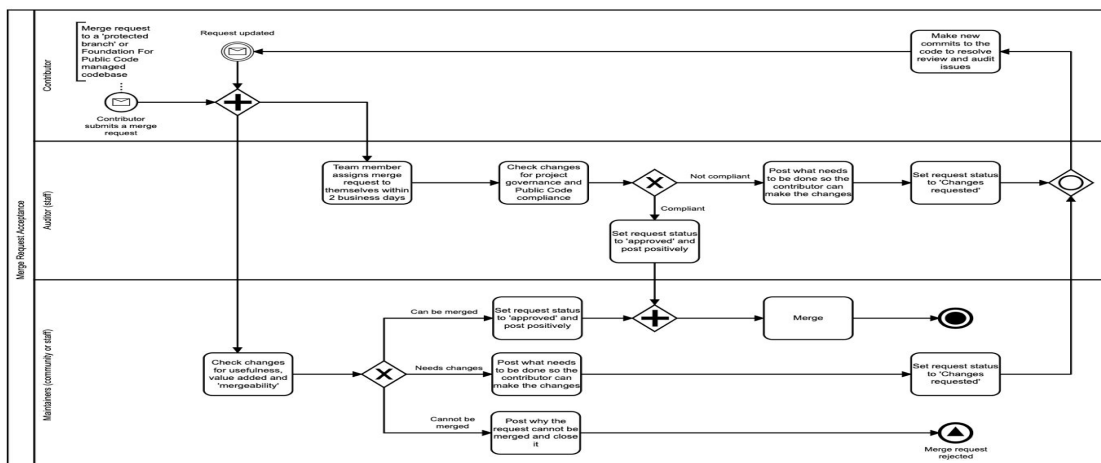
Die obligatorisch durchgeführte Prüfung über die Einhaltung der im folgenden dokumentierten Richtlinien soll die maschinellen Tests nur ergänzen, da Maschinen hervorragend geeignet sind, Dinge wie die Syntax zu überprüfen, nicht jedoch die Übereinstimmung der Ergebnisse mit den Erwartungen zu testen. Dinge, die für Menschen gedacht sind, wie z.B. das Testen, ob die Dokumentation im Kontext tatsächlich verständlich und zugänglich ist, ob die "Commits" Sinn machen und ob die Richtlinien der Gemeinschaft befolgt werden, sind für Maschinen (noch) nicht testbar.

Bei der Prüfung wird die gesamte Codebase, einschließlich Quellcode, Richtlinien, Dokumentation und Konversation, auf die Einhaltung sowohl der allgemeinen als auch der in der Codebase selbst festgelegten Standards geprüft.

### Wie der Prozess funktioniert bei FPC<sup>14</sup>

Jedes Mal, wenn ein Beitrag zu einer Codebase vorgeschlagen wird - zum Beispiel durch eine Merge-Anfrage - werden die FPC [Codebase-Stewards](#) den Beitrag auf die Einhaltung des Standards für den öffentlichen Code prüfen. Neue Beiträge können erst dann in die Codebase aufgenommen werden, wenn sie als konform mit dem Standard for Public Code genehmigt und von einem anderen Teilnehmer geprüft wurden.

Die Prüfung wird als eine Überprüfung des Beitrags dargestellt. Der Steward der Codebase gibt Zeile für Zeile Feedback und gibt Hinweise auf die Einhaltung der Vorschriften, um den Beitragenden bei der Verbesserung seines Beitrags zu unterstützen. Der Antrag auf Zusammenführung kann erst erfüllt werden, wenn die Codebase Stewards den Beitrag genehmigt haben.



## Zertifizierung einer gesamten Codebase im Vergleich zu einem einzelnen Beitrag

Damit die Codebase vollständig zertifiziert werden kann, muss jede sinnvolle Codezeile und die hinter dem Code stehenden Commits dem Standard entsprechen.

Wenn Codebases ab der ersten Merge-Anforderung vollständig geprüft wurden, können sie sofort als konform mit dem Standard for Public Code zertifiziert werden.

Wenn der Prüfungsprozess zu einer bestehenden Codebase hinzugefügt wird, können die neuen Merge-Anforderungen zertifiziert werden, aber der bestehende Code kann nicht zertifiziert werden. Durch die Prüfung jeder neuen Merge-Anforderung kann die Codebase schrittweise in Richtung einer vollständigen Zertifizierung verschoben werden.

## Codebase Hosting

Die Codebase Deutschland soll bereits im Sommer 2020 als einheitlicher Ort mit einem GitLab Clone veröffentlicht werden. Dies hat den Vorteil, daß spezielle Funktionen zur Filterung und Suche, Kategorisierung und Relevanz, die praktische Handhabung erleichtern. Am Anfang werden dort "nur" Forks zu sehen sein, im Laufe der ersten Jahre bei zunehmender Akzeptanz wird sich das voraussichtlich schnell ändern. Eine Sektion "international" wird das Auffinden von internationalen Projekten erleichtern und mit deutscher Verschlagwortung versehen.

## Die Ziele des Standard für Public Code

Dieser Standard unterstützt Entwickler, Designer, Manager und politische Entscheidungsträger dabei:

- qualitativ hochwertige Software und Richtlinien für bessere öffentliche Dienstleistungen zu entwickeln
- Codebasen zu entwickeln, die kontextübergreifend wiederverwendet und gemeinschaftlich gepflegt werden können
- technische Schuld und die Misserfolgsrate von Projekten zu reduzieren
- eine genauere Kontrolle über IT-Systeme zu haben und die Entscheidungskompetenz hier zu vergrößern
- Beziehungen zu Anbietern durch ein besseres Wirtschaftsmodell zu optimieren

FPC hilft allen Organisationen dabei, Open-Source-Software zur Verfügung zu stellen sowie einzurichten, nachhaltige Entwickler-Communities aufzubauen und ein vielfältiges Ökosystem für öffentlichen Code zu schaffen. Dies geschieht durch Codebase-Stewardship. Für diesen Prozess verwenden die Codebase Stewards den Standard für öffentlichen Code, um sicherzustellen, dass der von ihnen verwaltete Code von hoher Qualität ist und gemeinsam gewartet werden kann.

Potentielle Nutzer von Codebases, die anhand des Standard for Public Code getestet wurden, können erwarten, daß diese in hohem Maße wiederverwendbar, leicht zu warten und von hoher Qualität sind.

Der Standard for Public Code tut dies durch:

- die Festlegung einer gemeinsamen Terminologie für die Programmierung von öffentlichem Code
- Maßnahmen zur Unterstützung der Entwicklung eines qualitativ hochwertigen öffentlichen Code
- die Bereitstellung von Leitlinien zur Kriterienerfüllung und zur Operationalisierung der Einhaltung

Der Standard für das öffentliche Gesetzbuch soll zeit- und technologieunabhängig sein.

Der Standard for Public Code richtet sich an die Personen, die öffentlichen Code erstellen und wiederverwenden:

- politische Entscheidungsträger
- Geschäfts- und Projektmanager
- Entwickler und Designer

Diese Personen arbeiten bei:

- öffentlichen Einrichtungen: Institutionen und Verwaltungen
- Beratungsfirmen und Anbieter von Informationstechnologie und politischen Dienstleistungen für öffentliche Einrichtungen

Der Standard for Public Code richtet sich nicht an die Endnutzer öffentlicher Einrichtungen (Einwohner oder Bürger), Journalisten oder Wissenschaftler.

Der Standard ist ein lebendes Dokument. Lesen Sie den [leitfaden für Entwickler](#), um zu erfahren, wie Sie ihn weiterentwickeln und verbessern können.

## Leitfaden für Leser

Die Kriterien des Standards für Public Code haben einheitliche Abschnitte, die zusammen verdeutlichen, wie man guten öffentlichen Code erstellt.

### Anforderungen

Dieser Abschnitt listet auf, was zu tun ist, um den Standard zu erfüllen.

Um Mehrdeutigkeit einzuschränken, sind die Schlüsselwörter "MUSS", "DARF NICHT", "ERFORDERLICH", "SOLL", "SOLL NICHT", "SOLLTE", "SOLLTE NICHT", "EMPFOHLEN", "KANN" und "OPTIONAL" in diesem Dokument wie in [IETF RFC 2119](#) beschrieben zu interpretieren.

### Warum dies wichtig ist

Dieser Abschnitt erklärt, warum es für die Benutzer und Mitwirkenden dieser Codebase wichtig ist, dass diese Anforderungen eingehalten werden.

### Was dies nicht tut

In diesem Abschnitt wird die die Erwartungen gedämpft, indem erklärt wird, wovor das Folgen der Anforderungen Sie nicht bewahren wird.

Dies hilft:

- bei der korrekten Anwendung der Standards
- sicherzustellen, dass keine unerwarteten Probleme auftauchen

### Wie man testet

Dieser Abschnitt bietet Ihnen Möglichkeiten an, mit denen Sie feststellen können, ob ein Beitrag mit dem Standard übereinstimmt. Dies ist entscheidend, wenn Sie den Standard operationalisieren wollen.

Wir haben versucht, es so zu formulieren, dass auch jemand, der mit dem Thema nicht vertraut ist, eine grundlegende Prüfung auf Übereinstimmung mit dem Standard durchführen kann.

## Politische Entscheidungsträger: was Sie tun müssen

In diesem Abschnitt wird versucht, sich gezielt an die politischen Entscheidungsträger zu wenden, indem ihnen konkrete Maßnahmen angeboten werden, die sie in ihrer Rolle durchführen können.

Die politischen Entscheidungsträger legen die Prioritäten und Ziele der Projekte fest und sind möglicherweise weniger technisch versiert.

## Management: was Sie tun müssen

In diesem Abschnitt wird versucht, sich gezielt an das Managementteam zu wenden, indem konkrete Maßnahmen angeboten werden, die es in seiner Rolle durchführen kann.

Das Management ist verantwortlich für die pünktliche Projektdurchführung, das Management der Interessengruppen und die kontinuierliche Erbringung der Dienstleistung. Dabei sind sie sowohl von den politischen Entscheidungsträgern als auch von den Entwicklern und Designern vollkommen abhängig. Sie müssen die richtige Kultur schaffen, die richtigen Ressourcen aufstellen und die richtigen Strukturen zur Verfügung stellen, um großartige Dienstleistungen zu erbringen.

## Entwickler und Designer: was Sie tun müssen

Dieser Abschnitt versucht, gezielt mit Entwicklern und Designern zu sprechen, indem er ihnen konkrete Maßnahmen anbietet, die sie in ihrer Rolle durchführen können.

Die Entwickler sind in der Regel technisch besser ausgerichtet und haben mehr Einfluss auf die Bereitstellung von Dienstleistungen als die vorherigen Gruppen.

## Weiterführende Literatur

Lesen Sie mehr darüber, warum es diese Anforderungen gibt und wie sie erfüllt werden können.



## Glossar

### Code

Jedes explizit beschriebene Regelsystem. Dazu gehören sowohl Zivilgesetze - wie Gesetze, Richtlinien und Verordnungen - als auch der Quellcode, der zur Erstellung von Software verwendet wird. Beides sind Regeln, von denen einige von Menschen und andere von Maschinen ausgeführt werden.

### Codebase

Jedes einzelne Paket von Code (sowohl Quellcode als auch Richtlinie), die Tests und die Dokumentation, die zur Implementierung einer Richtlinie oder Software erforderlich sind.

Dies kann zum Beispiel ein Dokument oder ein Versionskontroll-Repository sein.

### Allgemeine Öffentlichkeit

Die breite Öffentlichkeit: Endbenutzer des Code und der darauf basierenden Dienste.

Die Bewohner einer Stadt werden beispielsweise als Endnutzer der Dienstleistungen einer Stadt und des gesamten Codes, der diese Dienstleistungen regelt, betrachtet.

### Offene Quelle

Open Source wird von der Open Source Initiative (odi.com) in ihrer Open Source Definition definiert.

### Offener Standard

Ein offener Standard ist jeder Standard, der die Anforderungen der Open-Source-Initiative an einen offenen Standard erfüllt.

### Öffentlicher Code

Öffentlicher Code ist sowohl bürgerlicher Code (wie Gesetze, Verordnungen, Verfahrens- oder Dienstanweisungen, Richtlinien oder Vorschriften) als auch Computer-Quellcode (wie Software und Algorithmen), der in einem öffentlichen Kontext, von Menschen oder Maschinen, ausgeführt wird.

Da öffentlicher Code dem öffentlichen Interesse dient, sollte er offen, lesbar, maschinenverstehbar, rechenschaftspflichtig, zugänglich und nachhaltig sein.

Durch die Entwicklung von öffentlichem Code, der unabhängig von dem lokalen Kontext, für den er entwickelt wurde, aber dennoch in diesem implementiert werden kann, sowie durch

die offene Dokumentation des Entwicklungsprozesses, kann öffentlicher Code einen Baustein für andere darstellen:

- die Neuimplementierung in ihrem lokalen Kontext
- als Ausgangspunkt für die weitere Entwicklung nehmen
- als Grundlage für das Lernen verwenden

Um die Wiederverwendung zu erleichtern, sollte öffentlicher Code entweder in die öffentliche Domäne freigegeben oder mit einer offenen Lizenz lizenziert werden, die es anderen erlaubt, das Werk frei anzuschauen, wiederzuverwenden und davon abgeleitete Werke zu produzieren.

### Zitervorschlag

XX:SfPC wobei sich XX auf die Ziffer der “Grundregeln” bezieht

### Lizenz

Das gesamte Werk steht unter der Creative Commons Lizenz CC0 1.0 und darf frei weiterverwendet werden.

# 1: Offener Code

## Anforderungen

- Der gesamte Quellcode aller verwendeten politischen Richtlinien und Software (sofern er nicht ausdrücklich gesperrt ist, z.B. zur Betrugserkennung verwendet wird) MUSS veröffentlicht und öffentlich zugänglich sein (Open by Default)
- Mitwirkende DÜRFEN KEINE sensiblen Informationen über Benutzer, ihre Organisation oder Dritte in das Repository hochladen. Beispiele für vertrauliche Informationen sind Konfigurationen, Benutzernamen und Passwörter, öffentliche Schlüssel und andere echte Zugangsdaten, die im Produktionssystem verwendet werden, die DSGVO ist zu beachten.
- Jeglicher derzeit nicht verwendeter Quellcode (wie neue Versionen, Vorschläge oder ältere Versionen) SOLLTE veröffentlicht werden.
- Der Quellcode KANN der allgemeinen Öffentlichkeit einen Einblick in den Quellcode oder die Richtlinien geben, die der spezifischen Interaktion mit Ihrer Organisation zugrunde liegen.

## Warum dies wichtig ist

### Offener Code:

- verbessert die Transparenz,
- erhöht die Code-Qualität,
- erleichtert die Auditierungsprozesse.

## Was dies nicht tut

- Den Quellcode oder die Richtlinie wiederverwendbar machen
- Die Codebase und den darin enthaltenen Code für möglichst viele Menschen verständlich machen

## Wie man testet

Die Quelle für jede derzeit verwendete Version wird im Internet veröffentlicht, wo sie eingesehen werden kann:

- von außerhalb der ursprünglich beitragenden Organisation
- ohne die Notwendigkeit irgendeiner Form von Authentifizierung oder Autorisierung.

## Entscheidungsträger: was Sie tun müssen

- Entwickeln Sie Richtlinien in der Öffentlichkeit.
- Offene und transparente politische Richtliniendiskussion ist zu priorisieren.

## Management: was Sie tun müssen

- Eine Kultur zu entwickeln, die Offenheit, Lernen und Feedback umfasst.
- Offene Zusammenarbeit mit externen Anbietern, StartUps und Freiberuflern

## Entwickler und Designer: was Sie tun müssen

- Klare Trennung von Daten und Code, um die oben genannte Anforderung bezüglich sensibler Informationen zu erfüllen.

## Weiterführende Literatur

[Coding in the Open](#), UK Government Digital Service.

[When code should be open or closed](#), UK Government Digital Service.

[Security considerations when coding in the open](#), UK Government Digital Service.

[Deploying software regularly](#), UK Government Digital Service.

[How GDS uses GitHub](#), UK Government Digital Service.

## 2: Bundle Policy und Quellcode

### Anforderungen

- Eine Codebase MUSS die Richtlinie enthalten, auf der der Quellcode basiert.
- Eine Codebase MUSS den gesamten Quellcode enthalten, auf dem die Richtlinie basiert.
- Alle Richtlinien und der Quellcode, auf denen die Codebase basiert, MÜSSEN dokumentiert, wiederverwendbar und portabel sein.
- Die Richtlinie SOLLTE in maschinenlesbaren und eindeutigen Formaten bereitgestellt werden.
- Kontinuierliche Integrationstests SOLLTEN bestätigen, dass der Quellcode und die Richtlinie kohärent ausgeführt werden.

### Warum dies wichtig ist

Dadurch wird sichergestellt, dass der Zugang sowohl zum Quellcode als auch zu den Richtlinien-Dokumenten gewährleistet ist, um die effektive Wiederverwendung einer Codebase zu erleichtern.

### Was dies nicht tut

- Garantieren, dass eine Codebase die Bundle Policy widerspiegelt.
- Stellen Sie sicher, dass die Pakete mit der lokalen technischen Infrastruktur oder dem rechtlichen Rahmen einer bestimmten öffentlichen Organisation übereinstimmen.

### Wie man testet

- Die Richtlinien werden in maschinen-verstehbaren und eindeutigen Formaten bereitgestellt.
- Kontinuierliche Integrationstests bestätigen, dass der Quellcode und die Richtlinie kohärent ausgeführt werden.

### Politische Entscheidungsträger: Was Sie tun müssen

- Arbeiten Sie mit Software-Entwicklern und den Designern der Richtlinien zusammen, um sicherzustellen, dass es keine Diskrepanz zwischen Richtliniencode und Quellcode gibt.
- Dokumentieren Sie Richtlinien in Formaten, die eindeutig und maschinenverstehbar sind, wie z.B. [Business Process Model and Notation](#), [Decision Model and Notation](#) und [Case Management Model Notation](#).
- Verfolgen Sie die Richtlinie mit [derselben Versionskontrolle](#) und Dokumentation, die zur Verfolgung des Quellcodes verwendet wird.

- Checken Sie regelmäßig, um zu verstehen, wie sich der nicht zur Richtlinie gehörende Code in der Codebase geändert hat und ob er noch den Absichten der Richtlinie entspricht.

### Management: was Sie tun müssen

- Halten Sie politische Entscheidungsträger, Entwickler und Designer während des gesamten Entwicklungsprozesses eingebunden und verbunden.
- Stellen Sie sicher, dass politische Entscheidungsträger, Entwickler und Designer auf die gleichen Ziele hinarbeiten.

### Entwickler und Designer: was Sie tun müssen

- Machen Sie sich mit der Notation der Prozessmodellierung vertraut, die die politischen Entscheidungsträger in Ihrer Organisation verwenden
- Arbeiten Sie mit den politischen Entscheidungsträgern zusammen, um sicherzustellen, dass es keine Diskrepanz zwischen dem Richtliniencode und dem Quellcode gibt.
- Geben Sie Feedback, wie die Dokumentation der Richtlinien klarer gestaltet werden kann.

### Weiterführende Literatur

[Free online tools for building BPMN, CMMN and DMN diagrams at bpmn.io](#), Camunda  
[BPMN Quick Guide](#), Trisotech

## 3: Erstellen Sie wiederverwendbaren und portablen Code

### Anforderungen

- Die Codebase MUSS so entwickelt werden, dass sie in verschiedenen Kontexten wiederverwendbar ist.
- Die Codebase MUSS unabhängig sein von geheimem, nicht offen gelegten, proprietären oder nicht offenem lizenzierten Code oder Services für die Ausführung und das Verständnis.
- Die Codebase MUSS von mehreren Parteien verwendet werden.
- Die Roadmap SOLLTE durch die Anforderungen mehrerer Parteien beeinflussbar sein.
- Der Code SOLLTE allgemein einsetzbar und konfigurierbar sein.
- Die Codebase SOLLTE eine [publiccode.yml](#)-Metadatenbeschreibung enthalten, so dass sie leicht auffindbar sind.
- Der Code und seine Dokumentation SOLLTE keine situationsspezifischen Informationen enthalten. Beispielsweise sollten persönliche und organisatorische Daten sowie Token und Passwörter, die im Produktionssystem verwendet werden, niemals enthalten sein.

### Warum dies wichtig ist

- Ermöglicht es anderen politischen Entscheidungsträgern, Entwicklern und Designern, das von Ihnen Programmiertere wiederzuverwenden, zu testen, zu verbessern und diese Verbesserungen wieder einzubringen, was zu besserer Qualität, günstigerer Wartung und höherer Zuverlässigkeit führt.
- Macht den Code für Neue einfacher zu verstehen (da er allgemeiner ist).
- Macht es einfacher, die Mission, die Vision und den Umfang der Codebase zu kontrollieren, da die Codebase durchdacht und zielgerichtet auf Wiederverwendbarkeit ausgelegt ist.
- Codebases, die von mehreren Parteien verwendet werden, haben einen Wert, der breit genug ist, um als öffentliches Gesetzbuch zu gelten, und profitieren eher von einer sich selbst tragenden Gemeinschaft.

### Was dies nicht tut

- Andere dazu zu bringen, die Codebase wiederzuverwenden.
- Eine Gemeinschaft aufzubauen.
- Verlagerung der Verantwortung für Dokumentation, Unterstützung, Fehlerbehebung usw. auf eine andere Partei.

## Wie man testet

- Fragen Sie jemanden in einer ähnlichen Rolle bei einer anderen Organisation, ob er Ihre Codebase wiederverwenden könnte und was das bedeuten würde.
- Die Codebase wird von mehreren Parteien oder in verschiedenen Kontexten verwendet.

## Politische Entscheidungsträger: was Sie tun müssen

- Dokumentieren Sie Ihre Richtlinien so klar und detailliert, dass sie auch außerhalb ihres ursprünglichen Kontexts verstanden werden kann.
- Stellen Sie sicher, dass Ihre Organisation vom Projekt als bekannter Benutzer aufgeführt wird.

## Management: was Sie tun müssen

- Stellen Sie sicher, dass Interessenvertreter und Geschäftsinhaber verstehen, dass die Wiederverwendbarkeit ein explizites Ziel des Projekts ist, da sie die technische Schuld reduziert und die Nachhaltigkeit der Codebase gewährleistet.

## Entwickler und Designer: was Sie tun müssen

- Die Quelle sollte für die Wiederverwendung durch andere Benutzer und Organisationen ausgelegt sein.
- Die Quelle sollte so gestaltet sein, dass sie ein allgemeines Problem anstelle eines spezifischen Problems löst.
- Jemand in einer ähnlichen Organisation, der mit einem ähnlichen Problem konfrontiert ist, sollte in der Lage sein, Ihre Lösung zu verwenden.

## Weiterführende Literatur

Making source code open and reusable, UK Government Digital Service DevOps Handbuch. (Kim Gene)

[https://en.wikipedia.org/wiki/Twelve-Factor\\_App\\_methodology](https://en.wikipedia.org/wiki/Twelve-Factor_App_methodology)

<https://de.wikipedia.org/wiki/Architekturmuster>



## 4: Beiträge willkommen heißen

### Anforderungen

- Die Codebase MUSS einen öffentlichen Issue Tracker haben, der Vorschläge von jedermann akzeptiert.
- Die Codebase MUSS es jedem ermöglichen, Vorschläge für Änderungen an der Codebase einzureichen.
- Die Dokumentation MUSS einen Link sowohl zum öffentlichen Issue Tracker als auch zu den eingereichten Codebase-Änderungen enthalten, z.B. in einer README-Datei.
- Die Codebase MUSS eine E-Mail-Adresse für Sicherheitsfragen und die verantwortungsvolle Offenlegung enthalten.
- Die Codebase MUSS Richtlinien für Beiträge enthalten, in denen erläutert wird, wie sich die Beitragenden beteiligen können, z.B. in einer CONTRIBUTING Datei.
- Das Projekt MUSS Kommunikationskanäle für Benutzer und Entwickler haben, z.B. E-Mail-Listen.
- Die Codebase MUSS eine öffentlich zugängliche Roadmap haben.
- Die Codebase SOLLTE für das engagierte Engagement der beteiligten Organisationen bei der Entwicklung und Wartung werben.
- Die Dokumentation SOLLTE Anweisungen enthalten, wie potenziell sicherheitsrelevante Probleme über einen geschlossenen Kanal gemeldet werden können.
- Die Codebase SOLLTE die Leitung der Codebase, die Beiträge und ihre Gemeinschaft dokumentieren, zum Beispiel in einer GOVERNANCE-Datei.
- Die Codebase KANN einen Verhaltenskodex für die Beitragenden enthalten.

### Warum dies wichtig ist

- Ermöglicht die gemeinsame Nutzung einer gemeinsamen digitalen Infrastruktur.
- Verhindert Verzweigungen von Codebases, in denen sich eine Gemeinschaft, die an einem Projekt arbeitet, aufteilt, weil es keinen gemeinsamen Fortschritt gibt.
- Hilft Benutzern bei der Entscheidung, eine Codebase statt einer anderen zu verwenden.
- Ermöglicht es Benutzern, Probleme zu beheben und der gemeinsamen Codebase Funktionen hinzuzufügen, was zu besserer, zuverlässigerer und funktionsreicherer Software führt.

### Was dies nicht tut

- Garantieren, dass andere die Codebase wiederverwenden werden.

## Wie man testet

- Es gibt einen Public Issue Tracker.
- Es ist möglich, Vorschläge für Änderungen an der Codebase einzureichen.
- Es ist möglich, an einer Diskussion mit anderen Benutzern über die Software teilzunehmen.
- Es gibt Richtlinien für Beiträge.

## Politische Entscheidungsträger: was Sie tun müssen

- Fügen Sie der Codebase eine Liste aller anderen Ressourcen hinzu, die Politikexperten, Nichtregierungsorganisationen und Akademiker zum Verständnis oder zur Wiederverwendung Ihrer Politik nützlich finden würden.
- Verfolgen Sie politische Fragen in der Codebase, so dass ein relevanter externer Politikexperte freiwillig helfen kann.
- Erwägen Sie, Kontaktdaten hinzuzufügen, damit andere politische Entscheidungsträger, die eine Wiederverwendung in Betracht ziehen, Sie um Rat fragen können.

## Management: was Sie tun müssen

- Vergewissern Sie sich, dass die Dokumentation erklärt, wie Ihre Organisation an dem Projekt beteiligt ist, welche Ressourcen ihr dafür zur Verfügung stehen und für wie lange
- Verfolgen Sie Managementprobleme in der Codebase, so dass externe Manager mit einschlägiger Erfahrung ehrenamtlich helfen können.
- Unterstützen Sie Ihre erfahrenen politischen Entscheidungsträger, Entwickler und Designer, damit sie so lange wie möglich zur Codebase beitragen können.

## Entwickler und Designer: was Sie tun müssen

- Reagieren Sie umgehend auf Anfragen.
- Informieren Sie Ihr Management über die Zeit und die Ressourcen, die Sie zur Unterstützung anderer Mitwirkender benötigen.

## Weiterführende Literatur

[Building welcoming communities](#), Open Source Guides

Der [Contributor Covenant](#) ist ein vorbildlicher Verhaltens-Code

[Leadership and governance](#) für wachsende Open-Source-Gemeinschaftsprojekte, von Open Source Guides.

[The benefits of coding in the open](#), UK Government Digital Service

[Building online communities](#), Pieter Hintjens

[Verdaccio's security policy](#) ist ein wirklich schönes Beispiel.

## 5: Versionskontrolle pflegen

### Anforderungen

- Sie MÜSSEN eine Möglichkeit haben, die Versionskontrolle für Ihren Code zu pflegen.
- Alle Dateien in einer Codebase MÜSSEN versionskontrolliert sein.
- Alle Entscheidungen MÜSSEN in Commit-Meldungen dokumentiert werden.
- Jede Commit-Nachricht MUSS, wo immer möglich, auf Diskussionen und Probleme verweisen.
- Sie SOLLTEN relevante Änderungen in Commits gruppieren.
- Sie SOLLTEN verschiedene Versionen Ihrer Codebase markieren, z. B. durch Revisions-Tags oder textuelle Labels.
- Sie SOLLTEN Dateiformate bevorzugen, die leicht versionskontrolliert werden können.

### Warum dies wichtig ist

Versionskontrolle bedeutet, Änderungen an Ihrem Code im Laufe der Zeit zu verfolgen. Dadurch können Sie eine strukturierte Dokumentation der Geschichte der Codebase erstellen. Dies ist für die Zusammenarbeit in großem Maßstab unerlässlich.

Die Versionskontrolle ermöglicht Ihnen dies:

- zu einer früheren Version der Codebase zurückzukehren, wann immer Sie wollen,
- zeichnen Sie Ihre Änderungen und die Gründe für die Änderungen auf, damit zukünftige Entwickler den Prozess besser verstehen können,
- zwei verschiedene Versionen zu vergleichen,
- parallel als Team an Veränderungen arbeiten, bevor sie zusammengeführt werden.

### Was dies nicht tut

- Ersatz dafür, die [Reife anzuzeigen](#)
- Die korrekte Ausführung Ihres Codes zu garantieren
- Mitwirkende zu garantieren.

### Wie man testet

- Die Codebase wird mit Hilfe von Software wie Git in der Versionskontrolle gehalten.
- Alle Commit-Nachrichten erklären:
  - warum die Änderung vorgenommen wurde,
  - was die Diskussion über die Änderung war oder wo man sie finden kann (mit einer URL).
- Es ist möglich, auf eine bestimmte Version der Codebase zuzugreifen, z.B. durch ein Revisionstag oder ein Text-Label.

## Politische Entscheidungsträger: was Sie tun müssen

- Wenn aufgrund einer Richtlinienänderung eine neue Version der Codebase erstellt wird, stellen Sie sicher, dass dies in der Dokumentation klar und deutlich angegeben wird:
  - was die Richtlinienänderung ist,
  - wie es die Codebase verändert hat.

Beispielsweise würde das Hinzufügen einer neuen Kategorie von Antragstellern zu einer Codebase, die die Erteilung von Genehmigungen verwaltet, als eine Änderung der Richtlinien betrachtet werden.

## Management: was Sie tun müssen

- Unterstützen Sie politische Entscheidungsträger, Entwickler und Designer, damit sie sich darüber im Klaren sind, welche Verbesserungen sie an der Codebase vornehmen - Verbesserungen vorzunehmen ist kein Risiko für die Öffentlichkeitsarbeit!

## Entwickler und Designer: was Sie tun müssen

- Schreiben Sie klare Commit-Nachrichten, so dass es leicht verständlich ist, warum der Commit gemacht wurde.
- Markieren Sie verschiedene Versionen, so dass es einfach ist, auf eine bestimmte Version zuzugreifen, z.B. mit Hilfe von Revisions-Tags oder textlichen Bezeichnungen.
- Schreiben Sie klare Commit-Nachrichten, damit Versionen sinnvoll verglichen werden können.
- Arbeiten Sie mit Entscheidungsträgern zusammen, um zu beschreiben, wie der Quellcode nach einer Änderung der Richtlinie aktualisiert wurde.

## Weiterführende Literatur

[Producing OSS: Version Control Vocabulary](#), Karl Fogel.

[Maintaining version control in coding](#), UK Government Digital Service.

[Semantic Versioning Specification](#) - wird von vielen Codebases zur Kennzeichnung von Versionen verwendet.

## 6: Überprüfung der Beiträge verlangen

### Anforderungen

- Alle Beiträge, die akzeptiert oder zur Freigabe von Versionen der Codebase verpflichtet sind, MÜSSEN von einem anderen Mitwirkenden überprüft werden.
- Rezensionen MÜSSEN Quelle, Richtlinien, Tests und Dokumentation enthalten.
- Rezensenten MÜSSEN Rückmeldung zu allen getroffenen Entscheidungen und zur Umsetzung in der Rezension geben.
- Die Beiträge SOLLTEN den in der Codebase festgelegten Standards, der Architektur und den Entscheidungen entsprechen, um die Überprüfung zu bestehen.
- Die Überprüfung SOLLTE die Ausführung sowohl des Codes als auch der Tests der Codebase umfassen.
- Beiträge SOLLTEN von jemandem überprüft werden, der in einem anderen Kontext als der Beitragende steht.
- Versionskontrollsysteme SOLLTEN keine nicht begutachteten Beiträge in Freigabeversionen akzeptieren.
- Überprüfungen SOLLTEN innerhalb von zwei Werktagen erfolgen.
- Rezensionen KÖNNEN von mehreren Rezensenten durchgeführt werden.

### Warum dies wichtig ist

- Erhöht die Qualität der Codebase.
- Verringert sowohl die Sicherheit als auch die operativen Risiken.
- Schafft eine Kultur, in der jeder Beitrag gut ist.
- Fängt die offensichtlichsten Fehler auf, die passieren können.
- Gibt den Beitragenden die Sicherheit, dass ihre Beiträge nur dann akzeptiert werden, wenn sie wirklich einen Mehrwert schaffen.
- Gewährt den Beitragenden eine garantierte Zeit für Feedback oder gemeinschaftliche Verbesserungen.

### Was dies nicht tut

- Die korrekte Lösung eines Problems zu garantieren.
- Zu bedeuten, dass die Rezensenten haftbar sind.
- Einen Beitragenden vom Schreiben von Dokumentation und Tests freizustellen.
- Die richtigen Gutachter zur Verfügung zu stellen.

### Wie man testet

- Jeder Commit in der History wurde von einem anderen Mitarbeiter in einem anderen Kontext überprüft.

## Politische Entscheidungsträger: was Sie tun müssen

- Einführung einer "Vier-Augen-Politik", bei der alles, nicht nur der Code, überprüft werden sollte.
- Verwenden Sie ein Versionskontrollsystem oder eine Methodik, die Überprüfung und Feedback ermöglicht.

## Management: was Sie tun müssen

- Machen Sie die Bereitstellung von sehr gutem Code zu einem gemeinsamen Ziel.
- Stellen Sie sicher, dass das Schreiben und die Überprüfung von Beiträgen zu Quelle, Richtlinien, Dokumentation und Tests als gleichwertig betrachtet werden.
- Schaffen Sie eine Kultur, in der alle Beiträge willkommen sind und jeder die Möglichkeit hat, sie zu überprüfen.
- Stellen Sie sicher, dass kein Beitragender jemals allein mit seinem Beitrag zu einem Projekt ist.

## Entwickler und Designer: was Sie tun müssen

- Bitten Sie andere Mitwirkende am Projekt, die Arbeit in Ihrer Organisation oder außerhalb des Projekts zu überprüfen.
- Versuchen Sie, so oft wie möglich auf die Anfragen anderer zur Überprüfung des Codes zu antworten.

## Weiterführende Literatur

[How to review code the GDS way](#), UK Government Digital Service.  
Branch protection auf [GitHub](#) und [GitLab](#).

## 7: Dokumentieren Sie Ihre Ziele

### Anforderungen

- Die Codebase MUSS eine Dokumentation ihrer Ziele enthalten - wie eine Missions- und Zielklärung - die für Designer und Entwickler verständlich ist, damit sie die Codebase nutzen oder zu ihr beitragen können.
- Die Codebase SOLLTE eine Dokumentation über ihre Ziele enthalten, die für politische Entscheidungsträger und das Management verständlich ist.
- Die Codebase KANN eine Dokumentation ihrer Ziele für die Allgemeinheit enthalten.

### Warum dies wichtig ist

Die Dokumentation Ihrer Ziele:

- bietet Menschen eine einfache Möglichkeit zu entscheiden, ob die Codebase jetzt oder in Zukunft für sie interessant ist,
- hilft Ihnen, die eigene Entwicklung einzuordnen und anderen Interessengruppen und Mitwirkenden klar zu vermitteln, wofür die Codebase gedacht ist.

### Was dies nicht tut

- Garantieren, dass die Codebase das/die erklärte(n) Ziel(e) erreicht.
- Beiträge zu Ihrer Codebase garantieren
- Verhindern, dass andere Codebasen versuchen, die gleichen Ziele zu erreichen.

### Wie Sie testen

- Es gibt einen Eintrag in der Codebase-Dokumentation für die Codebase-Ziele, die Mission oder das Ziel.

### Politische Entscheidungsträger: was Sie tun müssen

Fügen Sie die Richtlinien Ziele der Codebase-Dokumentation hinzu, z.B. in der README.

### Management: was Sie tun müssen

- Fügen Sie organisatorische und geschäftliche Ziele der Codebase-Dokumentation hinzu, z.B. im README.

### Entwickler und Designer: was Sie tun müssen

- Fügen Sie die Technologie- und Designziele der Codebase-Dokumentation hinzu, z.B. im README.

## Weiterführende Literatur

- How to write project objectives von Marek Hajduczenia



## 8: Verwenden Sie einfaches Englisch

### Anforderungen

- Der gesamte Code und die Dokumentation MÜSSEN auf Englisch sein.
- Jede Übersetzung MUSS auf dem neuesten Stand der englischen Version sein und umgekehrt.
- In der Codebase DARF es keine Akronyme, Abkürzungen, Wortspiele oder juristische/domänenspezifische Begriffe ohne eine vorausgehende Erklärung oder einen Link zu einer Erklärung geben.
- Der Name des Projekts oder der Codebase SOLLTE beschreibend sein und frei von Akronymen, Abkürzungen, Wortspielen oder Branding.
- Die Dokumentation SOLLTE eine Lesestufe der unteren Sekundarstufe anstreben, wie in den Richtlinien für die Zugänglichkeit von Web-Inhalten 2 empfohlen.
- Jeder Code, jede Dokumentation und jeder Test KÖNNEN eine Übersetzung haben.

### Warum dies wichtig ist

- Macht die Codebase und ihre Funktion für eine größere Vielfalt von Interessengruppen in verschiedenen Kontexten verständlich.
- Hilft bei der Auffindbarkeit Ihrer Codebase.
- Kann Ihnen helfen, die Kriterien der Europäischen Union zu erfüllen, die den allgemeinen Zugang zu Informationen des öffentlichen Sektors vorschreibt.

### Was dies nicht tut

- Die Erklärungen zur Funktionalität Ihrer Codebase verständlich zu machen.
- Den Fachjargon Ihrer Organisation ohne Erklärung verständlich zu machen.
- Sie daran zu hindern, die UX/UI in der Landessprache / Zielsprache zu machen

### Wie Sie testen

- Überprüfen Sie, ob die Übersetzungen und die englische Version den gleichen Inhalt haben.
- Stellen Sie sicher, dass keine unerklärlichen Akronyme, Abkürzungen, Wortspiele oder juristische/domänenspezifische Begriffe in der Dokumentation enthalten sind.
- Testen Sie die Dokumentation auf Grammatik mit Hilfe von "Grammarly".
- Testen Sie die Dokumentation mit dem Hemingway-Texteditor auf Lesbarkeit.
- Fragen Sie jemanden außerhalb Ihrer normalen Arbeitsumgebung, ob er Ihren Inhalt versteht (z.B. einen Entwickler, der an einem anderen Projekt arbeitet).

### Politische Entscheidungsträger: was Sie tun müssen

- Überprüfen Sie regelmäßig mit Führungskräften, Designern und Entwicklern, ob sie verstehen, was Sie liefern und wie Sie es dokumentieren.

## Management: Was Sie tun müssen

- Versuchen Sie, die Verwendung von Akronymen, Abkürzungen, Wortspielen oder juristischen/domänenspezifischen Begriffen in der internen Kommunikation in und zwischen Teams und Interessengruppen einzuschränken.
- Seien Sie kritisch gegenüber der Dokumentation und den Beschreibungen in Vorschlägen und Änderungen - wenn Sie etwas nicht verstehen, werden andere wahrscheinlich auch damit Schwierigkeiten haben.

## Entwickler und Designer: was Sie tun müssen

- Überprüfen Sie regelmäßig mit politischen Entscheidungsträgern und dem Management, ob sie verstehen, was Sie liefern und wie Sie es dokumentieren.

## Weiterführende Literatur

- [Web Content Accessibility Guidelines 2.1, Guideline 3.1 Readable](#) - Textinhalte lesbar und verständlich machen.
- [Upgoer 5 text editor](#) - erlaubt nur die 1000 häufigsten Wörter.

## 9: Offene Standards verwenden

### Anforderungen

- Für die Features einer Codebase, die den Datenaustausch erleichtern, MUSS die Codebase einen offenen Standard verwenden, der den [Open Source Initiative Open Standard Requirements](#) entspricht.
- Wenn kein vorhandener offener Standard verfügbar ist, MUSS ein solcher entwickelt werden.
- Standards, die maschinell testbar sind, SOLLTEN gegenüber solchen, die es nicht sind, bevorzugt werden.
- Eine Funktionalität, die Features aus einem nicht offenen Standard verwendet (der also nicht den [Open Source Initiative Open Standard Requirements](#) entspricht), KANN bei Bedarf bereitgestellt werden, aber nur zusätzlich zu konformen Funktionen.
- Alle nicht konformen Standards, die verwendet werden, MÜSSEN in der Dokumentation eindeutig festgehalten werden.
- Die Codebase SOLLTE eine Liste aller verwendeten Standards enthalten, mit Links zu den Stellen, an denen sie aufgeführt sind.

### Warum dies wichtig ist

- Schafft Interoperabilität zwischen Systemen.
- Reduziert die mögliche Bindung an Hersteller
- Garantiert den Zugriff auf entsprechendes Wissen, das für die Wiederverwendung und Beiträge zur Codebase erforderlich ist.

### Was dies nicht tut.

- Machen Sie den Umgang mit der Software verständlich.

### Wie Sie testen

- Die Dokumentation enthält eine Liste der Standards
- Die Standards, die für alle Funktionen verwendet werden, die Interoperabilität mit anderen Komponenten und Systemen bieten, sind im Internet frei und öffentlich verfügbar.

### Politische Entscheidungsträger: was Sie tun müssen

- Verboten Sie die Beschaffung von Technologie, die nicht den [Open Source Initiative Open Standard Requirements](#) entspricht.

## Management: was Sie tun müssen

- Erwägen Sie, eine Bewertung über die Einhaltung offener Standards in die Überprüfung des Kodex einzubeziehen.

## Entwickler und Designer: Was Sie tun müssen

- Fügen Sie kontinuierliche Integrationstests zur Einhaltung der Standards hinzu.

## Weiterführende Literatur

- [Open Standard Requirements](#) von der Open-Source-Initiative
- [Tools-to-find-open-standards](#) des [Open Data Institute](#)

## 10: Kontinuierliche Integration betreiben

### Anforderungen

- Alle Funktionen im Quellcode MÜSSEN automatisierte Tests haben.
- Beiträge MÜSSEN alle automatisierten Tests bestehen, bevor sie in die Codebase aufgenommen werden.
- Die Beiträge MÜSSEN von kleinem Umfang sein.
- Die Codebase MUSS aktive Mitwirkende haben.
- Der Quellcode-Test und die Erfassung der Dokumentation MÜSSEN überwacht werden.
- Richtlinien und Dokumentation KÖNNEN auf Konsistenz mit dem Quellcode und umgekehrt getestet werden.
- Richtlinien und Dokumentation KÖNNEN auf Stil und fehlerhafte Links getestet werden.

### Warum dies wichtig ist

- Kontinuierliche Integration
  - ermöglicht es Ihnen, Probleme mit Ihrer Codebase schnell zu identifizieren,
  - ermöglicht es, Risiken einzugehen und sich auf die Problemlösung zu konzentrieren und gleichzeitig den Stress für die Mitwirkenden zu minimieren,
  - senkt die Eintrittsbarrieren für neue Mitwirkende, indem es das nötige Verständnis für Änderungsvorschläge niedrig hält,
  - führt zu besser wartbarem Code,
  - beschleunigt den Entwicklungszyklus.
- Kleinere, regelmäßige Beiträge sind in der Regel leichter zu bewerten und mit geringerem Risiko verbunden als umfangreiche, gelegentliche Änderungen.
- Projekte in aktiver Entwicklung bieten zuverlässiger Möglichkeiten für Zusammenarbeit und Feedback.

### Was dies nicht tut

- Eine fehlertolerante Infrastruktur zu schaffen, die perfekt funktioniert und skalierbar ist.
- Aussagekräftige Tests zu erstellen
- Für Situationen des realen Lebens zu testen
- Zu garantieren, dass der Code genau das gleiche Ergebnis wie die Richtlinie liefert.

### Wie man testet

- Es sind Tests vorhanden.
- Tools zur Code-Abdeckung prüfen, ob die Abdeckung 100% des Codes beträgt.

- Beiträge werden erst dann in die Codebase aufgenommen, wenn alle Tests bestanden sind.
- Jeder, der an der Codebase arbeitet, integriert seine Arbeit mindestens einmal täglich.
- Es gibt Beiträge aus den letzten drei Monaten.

### Politische Entscheidungsträger: was Sie tun müssen

- Beziehen Sie sowohl das Management als auch die Entwickler und Designer so früh wie möglich in den Prozess ein und binden Sie sie während der gesamten Entwicklung Ihrer Politik ein.
- Stellen Sie sicher, dass auch automatisierte Tests für die Dokumentation der Richtlinie eingerichtet werden.
- Passen Sie die Richtliniendokumentation umgehend an, wenn ein Test fehlschlägt.
- Stellen Sie sicher, dass der Code alle Änderungen an der Richtlinie widerspiegelt (siehe Versionskontrolle pflegen).

### Management: was Sie tun müssen

- Stellen Sie sicher, dass Sie so schnell und oft wie möglich mit echten Endbenutzern testen.
- Nutzen Sie die Dienste von Beratern, die oft kleine Teile liefern, statt nur gelegentlich große Teile.
- Treiben sie bei wichtigen Fehlschlägen die Veröffentlichung von Vorfallberichten und die öffentliche Diskussion über das Gelernte voran

### Entwickler und Designer: was Sie tun müssen

- Helfen Sie dem Management und den politischen Entscheidungsträgern, ihre Beiträge zu testen, indem Sie beispielsweise ihre Beiträge auf gebrochene Links oder den Stil prüfen.
- Häufiges Deployment.

### Weiterführende Literatur

- [What is continuous integration](#) von Martin Fowler.
- [What is continuous delivery](#) von Jez Humble.
- [Use continuous delivery](#) vom UK Government Digital Service.
- [Quality assurance: testing your service regularly](#) vom UK Government Digital Service

## 11: Unter einer offenen Lizenz veröffentlichen

### Anforderungen

- Der gesamte Code und die Dokumentation MÜSSEN so lizenziert werden, dass sie frei wiederverwendbar, veränderbar und weiterverteilbar sind.
- Der Software-Quellcode MUSS unter einer von der [OSI genehmigten Open-Source-Lizenz](#) lizenziert werden.
- Der gesamte Code MUSS mit einer Lizenzdatei veröffentlicht werden.
- Alle Quellcode-Dateien in der Codebase MÜSSEN einen Copyright-Vermerk und einen Lizenz-Header enthalten.
- Codebases KÖNNEN mehrere Lizenzen für verschiedene Arten von Code und Dokumentation haben.
- Die Dokumentation KANN unter [Creative-Commons-Lizenzen](#) veröffentlicht werden, die NICHT 'keine Derivate' oder 'nicht kommerziell' sind.

### Warum dies wichtig ist

- Ermöglicht es jedem, den Code zu sehen und zu verstehen, dass er ihn wiederverwenden kann, und wie er ihn wiederverwenden kann.

### Was dies nicht tut

- Die Verwendung des Codes durch bestimmte Akteure zu verhindern

### Wie man testet

- Es ist mindestens 1 Lizenzdatei in der Codebase vorhanden, die normalerweise als Lizenz bezeichnet wird.
- Wenn sich Quellcode in der Codebase befindet, ist eine Lizenz vorhanden, die auf der Liste der [von der OSI genehmigten Open-Source-Lizenzen](#) steht.

### Politische Entscheidungsträger: was Sie tun müssen

- Entwickeln Sie Richtlinien, die einen quelloffenen Code erfordern.
- Entwickeln Sie Richtlinien, die von nicht quelloffenem Code und nicht quelloffener Technologie bei der Beschaffung abschrecken.

### Management: was Sie tun müssen

- Arbeiten Sie nur mit Open-Source-Anbietern zusammen, die ihren Code unter einer Open-Source-Lizenz veröffentlichen.

### Entwickler und Designer: Was Sie tun müssen

- Fügen Sie zu jeder neu erstellten Codebase eine neue Lizenzdatei hinzu.

- Fügen Sie einen Copyright-Vermerk und einen Lizenz-Header zu jeder neu erstellten Quellcode-Datei hinzu.

#### Weiterführende Literatur

- [Open source definition](#) durch die Open-Source-Initiative - alle Open-Source-Lizenzen entsprechen dieser Definition.
- [Animated video introduction to Creative Commons](#) von Creative Commons Aotearoa New Zealand.
- [REUSE Initiative specification](#) für eindeutige, menschen- und maschinenlesbare Urheberrechts- und Lizenzinformationen.



## 12: Einen zusammenhängenden Stil verwenden

### Anforderungen

- Die Beiträge MÜSSEN entweder einem Kodierungs- oder Schreib-Styleguide entsprechen, entweder Ihrem eigenen oder einem bereits bestehenden, der in der Codebase oder einem Teil davon beworben wird.
- Die Beiträge MÜSSEN automatisierte Stiltests bestehen.
- Ihre Codebase SOLLTE inline-Kommentare und Dokumentation für nicht-triviale Abschnitte enthalten.
- Sie KÖNNEN Abschnitte in Ihren Styleguide in [verständlichem Englisch](#) einfügen.

### Warum dies wichtig ist

- Ermöglicht es Mitarbeitern in verschiedenen Umgebungen, gemeinsam an einem einheitlichen Produkt zu arbeiten.
- Die Vereinheitlichung des Vokabulars verringert die Reibung in der Kommunikation zwischen den Beteiligten.

### Was dies nicht tut

- Den Beitragenden zu helfen, gut zu schreiben oder effektiv zu erklären, was sie tun.

### Wie man testet

- Prüfen Sie, ob die Beiträge mit den in der Dokumentation angegebenen Styleguides übereinstimmen.

### Politische Entscheidungsträger: was Sie tun müssen

- Erstellen Sie einen Styleguide für Richtlinien und Dokumentation, befolgen Sie ihn und verbessern Sie ihn kontinuierlich. Dokumentieren Sie dies in der Codebase, z.B. im CONTRIBUTING oder README.

### Management: was Sie tun müssen

- Beziehen Sie schriftliche Sprach-, Quellen-, Test- und Richtlinienstandards in Ihre organisatorische Definition von Qualität ein.

### Entwickler und Designer: was Sie tun müssen

- Erstellen Sie einen Styleguide für Quellcode, Tests und Dokumentation, befolgen Sie ihn und verbessern Sie ihn kontinuierlich, und dokumentieren Sie ihn in der Codebase, z.B. im CONTRIBUTING oder README.
- Verwenden Sie einen Linter.
- Fügen Sie der Codebase Linter-Konfigurationen hinzu.

## Weiterführende Literatur

- [List of linters](#) von Hugo Martins.
- [Programming style](#) auf Wikipedia.
- <https://www.tiobe.com/> Ansatz staatlicher Norm für öffentlichen Code

## 13: Achten Sie auf die Reifegrad der Codebase

### Anforderungen

- Eine Codebase MUSS versioniert werden.
- Eine einsatzbereite Codebase DARF nur von anderen Codebasen abhängen, wenn diese ebenfalls einsatzbereit sind.
- Eine Codebase, die noch nicht einsatzbereit ist, MUSS eine dieser Kennzeichnungen führen:
  - Prototyp - zum Testen des Look and Feel und zum internen Nachweis des Konzepts der technischen Möglichkeiten,
  - alpha - um mit einer begrenzten Anzahl von Benutzern geführte Tests zu veranstalten,
  - beta - um das Testen für einen weiten Teil der Öffentlichkeit zugänglich zu machen, z.B. um zu testen, ob die Codebase in großem Maßstab funktioniert,
  - Vorabversion - Code, der zur Veröffentlichung bereit ist, aber noch keine formale Genehmigung erhalten hat.
- Eine Codebase SOLLTE ein Protokoll der Änderungen von Version zu Version enthalten, zum Beispiel im CHANGELOG.

### Warum dies wichtig ist

- Die klare Signalisierung des Reifegrades einer Codebase hilft Anderen bei der Entscheidung, ob sie das Projekt wiederverwenden, in es investieren oder zu ihm beitragen sollen.

### Was dies nicht tut

- Garantieren, dass andere Ihren Code verwenden werden.

### Wie Sie testen

- Die Codebase verfügt über eine Versionierung-Strategie, die dokumentiert ist.
- Es ist klar, wo man die neueste Version erhält.
- Die Codebase hängt nicht von Codebases ab, die mit einem weniger ausgereiften Status gekennzeichnet sind.

### Politische Entscheidungsträger: was Sie tun müssen

- Bei der Entwicklung von Richtlinien sollte man sich darüber im Klaren sein, dass jeder entwickelte Code getestet und verbessert werden muss, bevor er in den Dienst gestellt werden kann.
- Berücksichtigen Sie Änderungen an der Richtlinie, insbesondere wenn sie neue Versionen des Quellcodes auslösen.

## Management: was Sie tun müssen

- Stellen Sie sicher, dass sich Services nur auf Codebasen von gleicher oder größerer Reife als der Service stützen. Verwenden Sie beispielsweise keine Beta-Codebase in einem Produktions-Service oder eine Prototyp-Codebase in einem Beta-Service.

## Entwickler und Designer: Was Sie tun müssen

- Fügen Sie jeder Schnittstelle eine auffällige Kopfzeile hinzu, die den Reifegrad des Codes anzeigt.
- Versionieren Sie alle Veröffentlichungen

## Weiterführende Literatur

- [Service Design and Delivery Process](#) der Australian Digital Transformation Agency.
- [Service Manual on Agile Delivery](#) vom UK Government Digital Service.
- [What are the Discovery, Alpha, Beta and Live stages in developing a service? \[Video 0'0"59\]](#) vom UK Government Digital Service.