Finanzdirektion
des Kantons Bern

Direction des finances
du canton de Berne

Amt für Informatik
und Organisation

Office d'informatique
et d'organisation

Wildhainweg 9
Postfach 6935
3001 Bern
Telefon 031 633 59 00
Telefax 031 633 59 99
www.be.ch/kaio
info.kaio@fin.be.ch

Source Code und Dokumentation

Checkliste

Bearbeitungs-Datum	14. April 2023
Version	1.5.1
Dokument Status	Freigegeben
Klassifizierung	Nicht klassifiziert
Autor	Steiner Nicolas, FIN-KAIO-BS-AS1
Dokumentnummer	#405471

Inhaltsverzeichnis

1 Zweck

Diese Checklisten richtet sich an ICT-Fachpersonen. Sie unterstützt diese bei der Prüfung der Dokumentation und des Quellcodes eines Projekts. Sie können damit sicherstellen, dass das Qualitätsniveau der Dokumentation ausreicht für die Veröffentlichung einer Anwendung als Open Source und ob der Quellcode sowie alle vom Projekt verwendeten Drittkomponenten keine rechtlichen Risiken darstellen.

Zusätzlich wird sichergestellt, dass die Dokumentation des Projekts sowohl die Enduser als auch die technischen Fachpersonen gleichermassen adressiert.

2 Checklisten

2.1 Checkliste zur Dokumentation

	Verweis auf Kapitel
☐ Die Dokumentation ist beim Source Code abgelegt	3.1
☐ Die Dokumentation spricht sowohl die Enduser als auch die technischen Fachpersonen an.	3.2
☐ Die Einstiegsdokumentation ist in einer Datei `README.md` abgelegt.	3.2
☐ Die Lizenz des Projekts ist in einer Datei `LICENSE` abgelegt. Der variable Teil der Lizenz-Vorlage wurde entsprechend angepasst.	3.2
☐ Die Datei `README.md` enthält ein Mission-Statement	3.3
☐ Die Datei `README.md` beinhaltet einen Hinweis darauf, dass es sich im Projekt um Open-Source-Software handelt und verlinkt auf die Datei `LICENSE`	3.4
☐ Die wichtigsten Funktionen des Projekts sind in der Datei `README.md` aufgeführt.	3.5
☐ Der aktuelle Stand der Entwicklung ist in der Datei `README.md` beschrieben.	3.6
☐ Für Applikationen existiert eine Demo-Instanz und die Demo-Instanz ist in der Datei `README.md` verlinkt.	3.7
☐ Die fachliche Ansprechperson für das Projekt ist benannt und in der Datei `README.md` angegeben.	3.8
☐ Der Eigentümer des Projekts, zum Beispiel ein für das Projekt gegründeter Verein, ist in der Datei `README.md` angegeben.	3.8
☐ Für Applikationen steht eine Installationsdokumentation zur Verfügung, die in der Datei `README.md` verlinkt ist.	3.9
☐ Die Datei `README.md` verweist auf die Richtlinien für Entwickler (Developer Guidelines).	3.10
☐ Die Datei `README.md` verweist auf die Dokumentation für Entwickler (Developer Documentation).	3.11
☐ Die «Developer Guidelines» sind in der Datei `CONTRIBUTING.md` abgelegt.	3.10
☐ Es ist ein Code of Conduct für das Projekt definiert, beim Source Code in einer Markdown-Datei abgelegt und im `CONTRIBUTING.md` verlinkt	3.10
☐ Die Developer-Guidelines enthalten einen Hinweis, dass Code-Reviews in Form von Pull-Requests durchgeführt werden.	3.10

☐ Die Developer-Guidelines enthalten einen Hinweis, dass bei der Änderung an den verwendeten Libraries die entsprechende Liste der Copyright-Notices bzw. Attributions aktualisiert werden muss.	3.10
☐ Die Developer-Guidelines verweisen auf die «Developer Documentation».	3.10
☐ Die «Developer Documentation» beschreibt die technischen Abhängigkeiten des Projekts und welche Werkzeuge für die Entwicklung des Projekts notwendig sind.	3.11
☐ Die formellen Regeln für den Quellcode des Projekts, zum Beispiel bezüglich der Formatierung des Quellcodes, sind in der «Developer Documentation» beschrieben.	3.11
☐ Die «Developer Documentation» beschreibt, wie der Quellcode des Projekts organisiert ist und wie Funktionen technisch getestet werden können.	3.11
☐ Die «Developer Documentation» beinhaltet eine Architekturskizze mit Grobbeschreibung der Architektur.	3.11
☐ Falls das Projekt eine Anwendung ist, dann beschreibt die «Developer Documentation», wie die Anwendung zu Entwicklungs- und Debugging-Zwecken gestartet werden kann und welche Infrastruktur-Komponenten dazu notwendig sind.	3.11

2.2 Checkliste zum Source Code

	Verweis auf Kapitel
☐ Der Source Code beinhaltet keine sensitiven, schützenswerten, oder vertraulichen Daten.	4.2
☐ Die Source Code History beinhaltet keine sensitiven, schützenswerten oder vertraulichen Daten.	4.1 4.2
☐ Die Entwicklerfirma wurde darauf hingewiesen, dass Namen und E-Mail-Adressen der Autoren des Quellcodes veröffentlicht werden, falls diese Informationen im Quellcode oder im Source-Code-Management-Repository enthalten sind.	4.2 4.3
☐ Die vom Projekt verwendeten Libraries sind kompatibel mit der Lizenz des Projekts.	4.4
☐ Das Projekt verwendet einen Package-Manager um seine Dependencies zu managen.	4.5
☐ Das Projekt deklariert seine Lizenz im entsprechenden Deskriptor des Package-Managers.	4.6
☐ Der Source Code enthält eine Datei `THIRD-PARTY-LICENSES.md`, in der für jede verwendete Library der Projekt-Name der Library, die Homepage der Library, der SPDX-Identifier der Lizenz der Library sowie ein Link auf die Lizenz der Library aufgeführt ist.	4.7
☐ Bei Applikationen steht ein «About»-Dialog zur Verfügung, der auf die Datei `THIRD-PARTY-LICENSES.md` verweist.	4.7
☐ Im Rahmen des Code-Review-Prozesses ist sichergestellt, dass die Datei `THIRD-PARTY-LICENSES.md` mit jeder Änderung an den Libraries aktualisiert wird.	4.7

3 Begleitende Dokumentation zur Checkliste «Dokumentation»

3.1 Dokumentation beim Quellcode

Die Dokumentation soll Teil des Quellcodes des Projekts sein, damit auch Änderungen an der Dokumentation revisionssicher aufbewahrt werden und die Dokumentation analog dem Projekt versioniert ist. Damit ist automatisch sichergestellt, dass die Dokumentation, sofern sie entsprechend gewartet wird, nicht vom Entwicklungsstand des Projekts divergiert und Dokumentation zu älteren Versionen des Projekts jederzeit abgerufen werden können.

Ausserdem gibt das Projekt dadurch Dritten die Möglichkeit, einfach Änderungen an der Dokumentation als Beitrag zurückzugeben.

Als Textformat soll Markdown als Auszeichnungssprache (Markup-Language) verwendet werden, da sie einfach zu schreiben, weit verbreitet und maschinenlesbar ist. Zusätzlich werden Dokumente in Markdown von allen gängigen Plattformen in einem gut lesbaren Zielformat dargestellt. Markdown-Dokumente können unabhängig von der verwendeten Plattform mit statischen Site-Generatoren wie Jekyll, MkDocs, Gatsby oder Sphinx in ein beliebiges Zielformat dargestellt werden, zum Beispiel mit den Corporate-Identity-Vorgaben einer Verwaltung.

3.2 Adressaten der Dokumentation und Struktur

Die Dokumentation des Projekts soll sowohl die Enduser als auch die technischen Fachpersonen ansprechen, sich also nicht nur auf die Technik fokussieren.

Als Einstiegsdokument dient dabei eine Datei `README.md`. Dateien mit diesem Namen werden von allen gängigen Plattformen als Einstiegspunkt für die Dokumentation erkannt und dargestellt.

`README.md` soll einen raschen Überblick über den Zweck, den Umfang, den Zustand, die Lizenz und die Zielgruppen des Projekts geben. Konkret soll `README.md` die folgenden Punkte umfassen:

- Mission Statement
- Verweis auf die Lizenz in einer Datei `LICENSE`
- Wichtigste Grundfunktionen des Projekts
- Stand der Entwicklung
- Verweis auf eine Demo-Instanz
- Angabe der fachlichen Ansprechpersonen sowie der Projekteigentümerin oder des Projekt-Eigentümers
- Verweis auf eine Installationsdokumentation
- Verweise auf die «Developer Guidelines» in einer Datei `CONTRIBUTING.md` und die «Developer Documentation»

Beispiel Codeblock 1 README.md

```
# Projekt-Name

Mission Statement
```

```
Projekt-Name ist Open-Source-Software, lizensiert unter der [BSD 3-Clause Li-
zenz](LICENSE).

Projekt-Name bietet die folgenden Funktionen:

* Funktion #1
* Funktion #2
* Funktion #3
* Funktion #4
* Funktion #5

Stand der Entwicklung

Eine Demo von Projekt-Name ist unter [https://demo.projekt-
name.ch](https://demo.projekt-name.ch) verfügbar.
Ein Login ist mit dem Benutzernamen `demo` und dem Passwort `demo` möglich.

Die fachliche Ansprechpartnerin / der fachliche Ansprechpartner für Projekt-Name
ist Vorname Nachname. Er kann per E-Mail unter info.kaio@fin.be.ch kontaktiert wer-
den.

## Projekt-Name verwenden

* [Installation von Projekt-Name](docs/install-instructions.md)

## An der Entwicklung von Projekt-Name mitarbeiten

Bitte lese die folgenden Dokumente durch, um an der Entwicklung von Projekt-Name
mitzuarbeiten:

* [Developer Guidelines](CONTRIBUTING.md)
* [Developer Documentation](docs/development.md)
```

3.3 Mission Statement

Das Mission Statement, beziehungsweise eine Kurzbeschreibung des Projekts, sollte einleitend am Anfang von `README.md` aufgeführt werden. Mit Hilfe des Mission Statements kann die Leserin oder der Leser rasch entscheiden, ob das Projekt interessant ist oder nicht. Das Mission Statement sollte konkret und vor allem kurz sein.

Für weitere Ausführungen zum Mission Statement (inklusive Beispiel) siehe Producing Open Source Software, Have a Clear Mission Statement.

3.4 Akzentuieren von Open Source und Ablage der Lizenz

Direkt nach dem Mission Statement soll klar und unmissverständlich beschrieben sein, dass das Projekt Open-Source-Software ist. Die vom Projekt verwendete Lizenz sollte mit dem entspre-

chenden SPDX-Identifier angegeben und auf den konkreten Lizenztext in der Datei `LICENSE` verlinkt werden. Die Lizenz-Vorlagen beinhalten in der Regel einen variablen Teil (zum Beispiel Name des Projekts, Copyright-Jahr, Name des Urhebers), der im `LICENSE`-File angepasst werden muss.

Die meisten Plattformen erkennen Lizenzen in einer Datei `LICENSE` und bieten übersichtlich Informationen zur Lizenz an, zum Beispiel eine kurze Zusammenfassung der Lizenzbedingungen.

Für weitere Ausführungen siehe: Producing Open Source Software, State That the Project is Free.

3.5 Wichtigste Funktionen des Projekts

Die wichtigsten Funktionen sollen als Teil der Datei `README.md` beschrieben werden und das Mission Statement ergänzen.

Für weitere Ausführungen siehe Producing Open Source Software, Features and Requirements List.

3.6 Aktueller Entwicklungsstand

Damit schnell ersichtlich ist, in welchem Zustand das Projekt ist, soll der aktuelle Entwicklungsstand in `README.md` dokumentiert werden. Für die Leserin oder den Leser ist dabei wichtig zu erfahren, ob das Projekt in einem ausgereiften Zustand oder am Beginn der Entwicklung ist, ob es aktiv gewartet wird und mit welcher Frequenz neue Versionen veröffentlich werden.

In diesem Abschnitt kann beschrieben werden, welche Unterstützung von Dritten für das Projekt aktuell am wichtigsten ist. Das kann zum Beispiel eine Entwicklerin oder ein Entwickler mit bestimmten technologischen Kenntnissen sein oder eine Person, die die Dokumentation des Projekts überarbeitet.

Weitere Ausführungen siehe Producing Open Source Software, Development Status.

3.7 Demo-Instanz

Bei Applikationen wird empfohlen, eine Demo-Instanz zur Verfügung zu stellen, damit die Applikation ohne Installationsaufwand ausprobiert werden kann. Je nach Art der Applikation kann die Demo-Instanz dabei die produktive Installation oder eine Teststufe sein. Wichtig ist, dass die Leserin oder der Leser möglichst selbständig und unmittelbar Zugriff auf die entsprechende Instanz erhalten kann.

3.8 Fachliche Ansprechperson und Projekt-Eigentümer

Die Datei `README.md` soll die primäre fachliche Ansprechperson benennen, damit andere interessierte Verwaltungen und Organisationen möglichst direkt Kontakt aufnehmen können.

Es sollen keine persönlichen E-Mail-Adressen verwendet werden. Falls in einer DIR/STA/JUS oder dem zuständigen Verein keine dafür geeignete Adresse vorhanden ist, kann für Projekte des Kantons Bern auch die generelle E-Mail-Adresse vom KAIO (info.kaio@fin.be.ch) verwendet werden.

Ebenfalls sollte in der Datei `README.md` kurz beschrieben werden, welche Organisation die Projekt-Eigentümerin oder der Projekt-Eigentümer ist, insbesondere wenn für das Projekt ein Verein gegründet wurde.

3.9 Installationsdokumentation

Für Applikationen sollte im `README.md` auf eine Installationsdokumentation verwiesen werden, die beschreibt, welche Anforderungen an Hard- und Software gestellt werden, welche Infrastrukturkomponenten (zum Beispiel Datenbanken) genutzt werden und wie die Applikation installiert werden kann.

3.10 «Developer Guidelines»

Der Einstiegspunkt der «Developer Guidelines» soll in einer Markdown-Datei `CONTRIBUTING.md` abgelegt werden, wobei das `CONTRIBUTING.md` im `README.md` verlinkt wird. Die «Developer Guidelines» sind Teil der weiterführenden Dokumentation für Entwicklerinnen und Entwickler, die zum Projekt beitragen möchten und fokussiert sich primär auf die Zusammenarbeit und Kommunikation innerhalb des Projekts und nicht auf die Technik.

Die «Developer Guidelines» soll dabei die folgenden Punkte beinhalten und gegebenenfalls auf die entsprechenden Dokumente verweisen:

- Ein Verweis auf den «Code of Conduct» des Projekts. Der «Code of Conduct» legt die erwarteten sozialen Normen innerhalb des Projekts fest.
- Es wird empfohlen, den Contributor Covenant Code of Conduct lizensiert unter CC-BY-4.0, als «Code of Conduct» zu wählen oder darauf aufzubauen.
- Ein Hinweis, dass das Projekt Code-Reviews in Form von GitHub Pull-Requests durchführt, mit einem Verweis auf die Pull-Request-Dokumentation von GitHub.

Abschliessend soll auf die «Developer Documentation» verwiesen werden.

Beispiel Codeblock 2 CONTRIBUTING.md

```
# An der Entwicklung von Projekt-Name mitarbeiten

Wir freuen uns auf deine Beiträge zu diesem Projekt und bitten dich folgende Punkte
zu beachten.

# Code-Reviews

Für alle Code-Änderungen, auch von Projekt-Mitgliedern, wird ein Code-Review durch-
geführt.
Dafür verwenden wir GitHub Pull-Requests. Lese die
[GitHub Hilfe](https://help.github.com/articles/about-pull-requests/) um mehr über
Pull-Requests zu erfahren.

# Code of Conduct

Bitte lese und beachte unseren [Code of Conduct](CODE-OF-CONDUCT.md).
```

Beispiel Codeblock 3 CODE-OF-CONDUCT.md (von Contributor Covenant Code of Conduct)

```
# Contributor Covenant Code of Conduct

## Our Pledge

In the interest of fostering an open and welcoming environment, we as
contributors and maintainers pledge to making participation in our project and
our community a harassment-free experience for everyone, regardless of age, body
size, disability, ethnicity, gender identity and expression, level of experience,
education, socio-economic status, nationality, personal appearance, race,
religion, or sexual identity and orientation.

## Our Standards

Examples of behavior that contributes to creating a positive environment
include:

* Using welcoming and inclusive language
* Being respectful of differing viewpoints and experiences
* Gracefully accepting constructive criticism
* Focusing on what is best for the community
* Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

* The use of sexualized language or imagery and unwelcome sexual attention or
  advances
* Trolling, insulting/derogatory comments, and personal or political attacks
* Public or private harassment
* Publishing others' private information, such as a physical or electronic
  address, without explicit permission
* Other conduct which could reasonably be considered inappropriate in a
  professional setting

## Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable
behavior and are expected to take appropriate and fair corrective action in
response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or
reject comments, commits, code, wiki edits, issues, and other contributions
that are not aligned to this Code of Conduct, or to ban temporarily or
permanently any contributor for other behaviors that they deem inappropriate,
threatening, offensive, or harmful.

## Scope

This Code of Conduct applies both within project spaces and in public spaces
```

```
when an individual is representing the project or its community. Examples of
representing a project or community include using an official project e-mail
address, posting via an official social media account, or acting as an appointed
representative at an online or offline event. Representation of a project may be
further defined and clarified by project maintainers.

## Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be
reported by contacting the project team at [INSERT EMAIL ADDRESS]. All
complaints will be reviewed and investigated and will result in a response that
is deemed necessary and appropriate to the circumstances. The project team is
obligated to maintain confidentiality with regard to the reporter of an incident.
Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good
faith may face temporary or permanent repercussions as determined by other
members of the project's leadership.

## Attribution

This Code of Conduct is adapted from the [Contributor Covenant][homepage], version
1.4,
available at https://www.contributor-covenant.org/version/1/4/code-of-conduct.html

[homepage]: https://www.contributor-covenant.org
```

3.11 Developer Documentation

Während die «Developer Guidelines» die Zusammenarbeit und sozialen Normen des Projekts beschreiben, fokussiert sich die «Developer Documentation» auf die technischen Aspekte, um an der Entwicklung des Projekts teilzunehmen. Sie soll mindestens die folgenden Punkte beschreiben:

- Welche technischen Abhängigkeiten hat das Projekt und welche Werkzeuge sind für die Entwicklung des Projekts notwendig.
- Welche formellen Regeln gelten für den Quellcode des Projekts, zum Beispiel bezüglich der Formatierung des Quellcodes.
- Wie ist der Quellcode des Projekts organisiert und wie können Funktionen technisch getestet werden.
- Eine Architekturskizze mit Grobbeschreibung der Architektur, möglichst inklusive einer Kontextabgrenzung und der ersten Ebene der Bausteinsicht gemäss arc42.
- Für Applikationen: Wie kann die Applikation zu Entwicklungs- oder Debugging-Zwecken gestartet werden und welche Infrastruktur-Komponenten sind dafür notwendig.

4 Begleitende Dokumentation zur Checkliste «Source Code»

4.1 Source Code History

Der Source Code eines Projekts ist in der Regel in einem Source-Code-Management-System (SCM) wie zum Beispiel GitHub abgelegt. Diese Systeme speichern nicht nur den aktuellen Zustand des Source Codes, sondern auch Änderungen, die im Laufe der Zeit am Source Code durchgeführt wurden. Dazu zählen insbesondere auch Löschungen aus dem Source Code, inklusive Löschungen von Dateien.

Zusätzlich enthält das SCM Meta-Informationen zu Änderungen am Source Code, wie zum Beispiel den Namen und die E-Mail-Adresse der Autorin bzw. des Autors oder PGP-Signaturen bei signierten Commits und Tags.

4.2 Sensitive, schützenswerte oder vertrauliche Daten

Der Source Code und die Source Code History des Projekts dürfen öffentlich verfügbare Daten, wie zum Beispiel ein Postleitzahlenverzeichnis, zufällig generierte Daten oder synthetische Daten enthalten.

Der Source Code und die Source Code History dürfen keine sensitiven, schützenswerten oder vertraulichen Daten bzw. Informationen beinhalten. Dazu zählen insbesondere:

- Benutzernamen und Passwörter oder andere Geheimnisse wie private Keys, Access-Token oder Zertifikate.
- Interne DNS-Namen, URLs, Hostnamen, IP-Adressen, Netzwerkstrukturen oder Laufwerksbezeichnungen.
- Anonymisierte oder pseudonymisierte Daten (da bei der Anonymisierung oder Pseudonymisierung von Daten eine Möglichkeit bestehen bleibt, die Anonymisierung oder Pseudonymisierung rückgängig zu machen)

Auch wenn der aktuelle Zustand des Source Codes frei von sensitiven, schützenswerten oder vertraulichen Daten bzw. Informationen ist, können solche Daten bzw. Informationen über die History des SCM-Systems abgefragt werden, wenn sie in der Vergangenheit Teil des Source Codes oder der SCM-Meta-Informationen waren. Eine Bereinigung des aktuellen Source-Code-Zustands entfernt diese Informationen also nicht vollumfänglich.

Im Zweifelsfall wird daher empfohlen, die Source Codes History nach einer Bereinigung des Source Codes vollständig zu entfernen, bevor der Source Code veröffentlich wird.

Die Namen und E-Mail-Adressen der Autorinnen und Autoren des Quellcodes sind in der Regel direkt im Quellcode oder in den SCM-Meta-Informationen vorhanden, zum Beispiel in GitHub Commits oder annotierten GitHub Tags. Aus rechtlicher Sicht stellt dies kein Problem dar, da die Entwicklerfirma die Verantwortung trägt, die Veröffentlichung solcher Informationen zu regeln. Es wird jedoch empfohlen, die Entwicklerfirma darauf aufmerksam zu machen, dass mit einer Veröffentlichung der Software potentiell auch die Namen und E-Mail-Adressen der Autorinnen und Autoren des Quellcodes veröffentlicht werden.

4.3 Entfernen von Autor-Informationen

Die History von GitHub kann so verändert werden, dass Informationen zu Autorinnen und Autoren unkenntlich gemacht werden. Dazu kann beispielsweise das folgende Skript verwendet werden: https://www.adamdehaven.com/blog/update-commit-history-author-information-for-git-repository/ /

4.4 Berücksichtigung der Lizenzen von Libraries

Praktisch jedes Projekt verwendet Libraries (Software-Bibliotheken, Abhängigkeiten, Dependencies) von Dritten, um auf allgemein verwendbare Funktionen zurückgreifen zu können – ohne diese Funktionen selbst zu programmieren. Meist sind diese Libraries wieder von anderen Libraries abhängig, was zu mehrstufigen Abhängigkeiten führt (transitive Dependencies). Je nach Programmiersprache sind selbst bei kleineren Projekten hunderte Dependencies die Regel.

Für jede dieser Dependencies muss sichergestellt sein, dass ihre Lizenz mit der Projektlizenz kompatibel ist. Besonders problematisch sind dabei Dependencies, die nicht unter einer Open-Source-Lizenz stehen, die keine Lizenz deklariert haben, oder die eine virale Lizenz verwenden, die nicht mit der Projektlizenz übereinstimmt. Einige Libraries stehen unter mehr als einer Lizenz und erlauben der Verwenderin oder dem Verwender eine Lizenz zu wählen.

In allen gängigen Programmiersprachen können die Dependencies des Projekts (inklusive transitive Dependencies) sowie die Lizenzen der Dependencies automatisch aufgelistet werden. Dazu zählen:

- Das Project Info Reports Plugin des Package-Managers Apache Maven für Java-Projekte, das über den Dependencies-Report alle verwendeten Libraries inklusive der deklarierten Lizenz in HTML-Form erstellt.
- Der NPM License Checker für den NPM Package-Manager von JavaScript-Projekten, der alle verwendeten Libraries inklusive der deklarierten Lizenz in verschiedenen Formaten, zum Beispiel CSV, ausgeben kann.
- Der License Finder von Pivotal, der unterschiedliche Package-Manager und Programmiersprachen unterstützt, beispielsweise auch Ruby Gems, Python Eggs oder Godeps.

4.4.1 Verwenden von Libraries mit proprietären Lizenzen

Libraries, die unter einer proprietären Lizenz stehen, sind in Open-Source-Projekten grundsätzlich problematisch und können in der Regel nicht verwendet werden, insbesondere wenn das Open-Source-Projekt unter einer viralen Lizenz wie der AGPL veröffentlicht wurde.

Bitte nehmen Sie zusammen mit dem Rechtsdienst des KAIO Kontakt zum Hersteller auf, um zu klären, ob und unter welchen Bedingungen die proprietäre Library in Ihrem Projekt eingesetzt werden kann.

4.4.2 Für Projekte unter den Lizenzen BSD 3-Clause, MIT oder Apache 2.0

Für Projekte, die unter den Lizenzen BSD 3-Clause, MIT oder Apache 2.0 lizensiert sind, können Libraries eingesetzt werden, die unter den folgenden Lizenzen stehen (Auswahl):

Apache 2.0, Apache 1.0, BSD 3-Clause, BSD 2-Clause, ISC, LGPL (gelinkt), MIT, MPL

Libraries, die unter den folgenden Lizenzen stehen, können nicht eingesetzt werden (Auswahl):

AGPL, BSD 4-Clause, EUPL, GPL, GPL-SIK, LGPL (wenn nicht gelinkt)

Ist die Lizenz einer Library nicht in den entsprechenden Listen enthalten, bitte den Rechtsdienst des KAIO zur Klärung kontaktieren.

4.4.3 Für Projekte unter der Lizenz EUPL

Für Projekte, die unter der Lizenz EUPL lizensiert sind, können Libraries eingesetzt werden, die unter den folgenden Lizenzen stehen (Auswahl):

Apache 2.0, BSD, EUPL, LGPL (gelinkt), MIT, MPL

Libraries, die unter den folgenden Lizenzen stehen, können nicht eingesetzt werden (Auswahl):

AGPL, GPL, LGPL (wenn nicht gelinkt)

Ist die Lizenz einer Library nicht in den entsprechenden Listen enthalten, bitte den Rechtsdienst des KAIO zur Klärung kontaktieren.

4.5 Verwenden von Package-Managern

Praktisch alle gängigen Programmiersprachen bieten Package-Manager an, um ihre Dependencies zu verwalten, zum Beispiel Apache Maven für Java-Projekte, NPM für JavaScript-Projekte oder NuGet für .NET-Projekte.

Bei der Verwendung eines Package-Managers sind die vom Projekt verwendeten Dependencies nicht mehr direkt Teil des Projekt-Source-Codes, zum Beispiel indem Source Code der Dependency in das Projekt kopiert wird oder die Dependencies in Binärform in das Projekt kopiert werden. Stattdessen bezieht das Projekt alle Dependencies über entsprechende Deklarationen des Package-Managers.

Dadurch ist sichergestellt, dass alle Dependencies des Projekts mit Hilfe des Package-Managers aufgelistet werden können und es wird vermieden, dass der Quellcode der Dependencies innerhalb des Projekts modifiziert wird, ohne die Modifikation an die ursprünglichen Autorinnen und Autoren der Dependency zurückzugeben.

4.6 Deklaration der Projekt-Lizenz im Package-Manager Deskriptor

Das Projekt sollte seine Lizenz im entsprechenden Deskriptor des Package-Managers mit dem SPDX-Identifier deklarieren, damit die Projekt-Lizenz über den Package-Manager ausgegeben werden kann. Dies ist insbesondere für Libraries wichtig, damit Verwenderinnen und Verwender der Library automatisch deren Lizenz über den Package-Manager abfragen können.

4.7 Attributions und Hinweise zum Urheberrecht

Viele Libraries stehen unter einer Lizenz, die von der Verwenderin oder vom Verwender der Library fordert, einen Hinweis zum Urheberrecht (Original Copyright Notice) aufzuführen, oder eine Anerkennung (Attribution) der Nutzung der Library zu geben.

Dazu zählen die folgenden Lizenzen (Auszug):

Apache 2.0, ASL 1.1 (Apache 1.1), BSD, BSD 3-Clause, Creative Commons “Attribution” (CC BY), MIT, ISC

Für weitere Lizenzen siehe Google Open Source Documentation, Licenses - Notice

Damit sind in der Praxis fast alle verwendeten Libraries betroffen. Da bereits in Projekten mit geringem Umfang die Anzahl der verwendeten Libraries sehr hoch ist, ist ein juristisch einwandfreier Hinweis zum Urheberrecht bzw. die entsprechend notwendige Attribution nur mit sehr grossem Aufwand möglich.

Es wird daher empfohlen, wie folgt vorzugehen:

- Eine Liste der verwendeten Libraries über die entsprechenden Mechanismen des Package-Managers oder Tools wie den NPM License Checker bzw. den License Finder von Pivotal erstellen.
- Die Liste manuell aufbereiten, damit sie die folgenden Informationen beinhaltet: Offizieller Name des Projekts der Library, Homepage des Projekts der Library, Verwendete Lizenz der Library als SPDX-Identifier mit einem Link auf den Lizenztext.
- Die Liste im Source Code des Projekts ablegen, zum Beispiel in einer Datei `THIRD-PARTY-LICENSES.md`.
- Bei Applikationen: Einen Link in einem «About» bzw. im Dialog «Über diese Applikation» einfügen, der auf das `THIRD-PARTY-LICENSES.md` verweist.
- Die Nutzung des About-Dialogs für diesen Zweck ist weit verbreitet, zum Beispiel bei Google Chrome (Im Dialog «About Chrome»).
- Bei Libraries: Im `README.md` auf das `THIRD-PARTY-LICENSES.md` verweisen.
- Als Teil des Code-Review-Prozesses forcieren, dass Änderungen an den Libraries im `THIRD-PARTY-LICENSES.md` nachgetragen werden.

Beispiel Codeblock 4 THIRD-PARTY-LICENSES.md

```
Dieses Projekt verwendet Open-Source-Software:

* Spring Boot, http://projects.spring.io/spring-boot/, lizensiert unter [Apache-
2.0](http://www.apache.org/licenses/LICENSE-2.0)
* caniuse-db, https://github.com/Fyrd/caniuse, lizensiert unter [CC-BY-
4.0](https://creativecommons.org/licenses/by/4.0/)
```

Quellen und Lizenz

Die Checkliste, die begleitende Dokumentation und die entsprechenden Vorlagen basieren in Teilen auf den folgenden Quellen:

- Google Open Source Docs von Google LLC, lizenziert unter CC-BY-4.0
- Producing Open Source Software, How to run a Successful Free Software Project von Karl Fogel, lizenziert unter CC-BY-SA-4.0

Das entsprechende öffentlich verfügbare Ergebnis steht aufgrund der verwendeten Quellen unter der offenen Lizenz CC-BY-SA-4.0.

Dokument – Protokoll

Dokumentnummer #405471

Autor Joos Thomas, FIN-KAIO-AP-SW

Änderungskontrolle

Version	Name	Datum	Bemerkungen
0.1	Ferdinand Hübner	27.02.2018	Erstellung
0.2	Mario Siegenthaler	14.03.2018	Ergänzungen und Überführung in Word
0.3	Thomas Joos	01.05.2018	Kleinere Ergänzungen und Anpassungen
0.4	Ferdinand Hübner	14.05.2018	CLAs und License-Header pro Datei entfernt. AGPL entfernt und EUPL stattdessen hinzugefügt. Klärung Hinweis zur Veröffentlichung von Namen und E-Mail-Adressen der Quellcode-Autoren. Einarbeitung Feedback Cornelia Nussberger
0.5-0.7	Thomas Joos	13.06.2018	Überarbeitungen
0.8	Thomas Joos	13.06.2018	Finalisierung
1.0	Thomas Joos	04.07.2018	Schlussversion nach Genehmigung PB
1.1	Thomas Joos	12.09.2018	Fehlerkorrekturen und kleinere Anpassungen, geschlechtsneutrale Anpassungen
1.2-1.3	Stefan Schneider	13.09.2018	Überarbeitung
1.4	Thomas Joos	18.09.2018	Finalisierung
1.5	Kélèfa Keita	21.03.2023	Aktualisierung
1.5.1	Nicolas Steiner	14.04.2023	Aktualisierung

Prüfung

Version	Stelle	Datum	Visum	Bemerkung
0.3	BVE	09.05.2018	C. Nussberger	Kleinere Bemerkungen und Anpassungen
0.8	Thomas	13.06.2018	Tjo	---
0.9	PB	02.07.2018	PB	Genehmigung Portfolioboard
1.2	Stab	13.09.2018	Ssc	Rechtschreibeprüfung

Freigabe

Version	Stelle	Datum	Visum	Bemerkung
1.0	Abtl. / FBL	25.06.2018	mwe / rae	---
0.9	PB	02.07.2018	PB	Freigabe durch Portfolioboard