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

Open-Source-Open Source Plattform

Leitfaden Community Gestaltung

Bearbeitungs-Datum	14. April 2023
Version	1.5.1
Dokument Status	Freigegeben
Klassifizierung	Nicht klassifiziert
Autor	Steiner Nicolas, FIN-KAIO-BS-AS1
Dokumentnummer	#405475

Inhaltsverzeichnis

1 Zweck

Dieser Leitfaden ist für ICT-Fachpersonen bestimmt, die für eine Applikation verantwortlich sind und diese als Open Source anbieten wollen. Sie finden hier wichtige Informationen für die Organisation und den Aufbau einer Community.

Welche Art der Community für eine Applikation am besten geeignet ist, hängt sehr stark vom fachlichen Umfeld, den möglichen Nutzerinnen und Nutzern sowie der Vision der veröffentlichenden Institution ab. Es besteht keine allgemeine nützliche Beschreibung, wie eine Community aufgebaut werden soll. Auf Basis dieses Dokuments wird stattdessen ein konkretes Community-Konzept für die jeweilige Applikation entwickelt.

Als zweites Element wird zur Förderung der Standardisierung ebenfalls ein Verzeichnis von bereits bestehenden Community-Konzepten als Teil dieses Dokuments geführt.

Dieses Dokument beinhaltet keine direkten Empfehlungen für die Organisation von Communities für Libraries. Für diese wird im Normalfall kein eigenes Konzept entwickelt, sondern sie verwenden die in der Repository-Vorlage definierten, simplen Standardprozesse.

2 Struktur des Community-Konzepts

Das zu erstellende Dokument soll der folgende Kapitelstruktur folgen. Je nach Art der spezifischen Community sind nicht alle Unterkapitel erforderlich.

1. Zielsetzung
    a. Der Applikation
    b. Der Community
2. Organisation
    a. Eigentümerin / Eigentümer des Codes
    b. Gremien
    c. Stimmrechte
3. Roadmap und Change-Prozess
4. Entwicklungsprozess
    a. Committer-Rechte
    b. Review-Prozess
5. Kostenteilung und Lieferanten
6. Vermarktung
7. Umgang mit externen Beiträgen
    a. Umgang mit gemeldeten Fehlern
    b. Umgang mit Anfragen
    c. Umgang mit Pull Requests
    d. Umgang mit Forks
8. Betrieb der Applikation

3 Grundsatzentscheide

Zu Beginn der Erarbeitung des Community-Konzepts sollen drei Grundsatzentscheide getroffen werden:

a) Wie wird das Produkt-Management gelöst und wer bestimmt die Roadmap?
b) Wie sollen die Lieferanten eingebunden werden?
c) Wie ist die Aufteilung der Kosten geregelt?

Aus den Antworten auf diese Fragen ergibt sich dann die grundsätzliche Form der Community.

Wir empfehlen, für die Ausgestaltung der Community vor allem auf die nähere Zukunft zu fokussieren; wenn also beispielsweise noch keine konkreten Interessentinnen oder Interessenten für die Applikation vorhanden sind, dann macht es meist wenig Sinn, eine Community mit einem eigenen Verein und komplexen Prozessen zur Erarbeitung der Roadmap zu definieren.

3.1 Produkt-Management

Je nach Art der Applikation, ihrer strategischen Bedeutung und ihrem Status im Lebenszyklus gibt es unterschiedliche Varianten, das Produkt-Management abzubilden. Insbesondere relevant für das Community-Konzept sind die Definition der fachlichen und technischen Roadmap sowie die Release-Prozesse.

Möglichkeiten:

a) Kanton Bern: Der Kanton will die Kontrolle behalten, er definiert alleine die Roadmap
b) Gemeinsam: Gemeinsam mit anderen Verwenderinnen und Verwendern
c) Offene Organisation: Lose Verbindungen, keine aktive Roadmap

3.1.1 Kanton Bern

Wenn die Applikation eine hohe strategische Bedeutung hat oder der Kanton Bern darauf angewiesen ist, seine Änderungen schnell implementieren zu können, dann kann es Sinn machen, die Kontrolle zu behalten. Der Kanton Bern entscheidet alleine, was wann in die Software aufgenommen wird.

Für mögliche andere Verwenderinnen und Verwender der Applikation bedeutet dies aber, dass sie fast gezwungen sind, eine eigene Version (fork) der Applikation zu verwenden. Ansonsten haben sie kaum die Möglichkeit, die für sie wichtigen Anpassungen und Erweiterungen umzusetzen. Dies spricht aber nicht gegen eine Veröffentlichung als Open Source, die Werkzeuge zur Code-Verwaltung[1] bieten weitgehende Unterstützung für solche Szenarien; Anpassungen und Fehlerkorrekturen können selektiv in beide Richtungen übernommen werden.

Eine Teilung der Kosten ist bei diesem Modell typischerweise nur schwer möglich.

3.1.2 Gemeinsam

Die Entscheide betreffend Roadmap und Prioritäten sollen mit den anderen Mitgliedern der Community abgestimmt und gemeinsam getroffen werden. Der Kanton Bern bestimmt als Mitglied der Community mit, hat aber nicht das letzte Wort.

[1] Beispielsweise „GitHub“

Damit eine gemeinsame Beschlussfassung funktionieren kann, müssen vorgängig Strukturen und Regeln bestimmt werden. Dabei wird auch festgelegt, wie die Kosten für die Umsetzung der gemeinsam beschlossenen Anpassungen untereinander aufgeteilt werden.

Mit diesem Modell entsteht eine einzige Version der Applikation, die dann von allen Mitgliedern genutzt wird. Dadurch sind die Gesamtkosten verglichen mit den anderen Varianten deutlich tiefer. Die Flexibilität der einzelnen Verwenderinnen und Verwender ist aber ebenfalls geringer, sie müssen ihre Anpassungswünsche vorgängig mit allen andern abstimmen; die Entscheidungsfindung ist langsamer und aufwändiger.

Dieses Modell eignet sich am besten für Applikationen, die gemeinsam mit klar definierten anderen Partnerinnen und Partnern (beispielsweise andere Kantone) weiterentwickelt werden sollen.

3.1.3 Offene Organisation

In diesem Modell werden die Prozesse und Strukturen nur lose definiert. Der Kanton behält zwar formell die Kontrolle, ist aber offen für Beiträge und Anpassungen.

Es wird keine oder nur eine sehr grobe Roadmap definiert, der Konsens wird informell und in Diskussionen direkt auf der Publikationsplattform hergestellt. Die Community selber ist eher dynamisch, Mitglieder können eine Zeit lang sehr aktiv sein und sich dann wieder zurückziehen.

Dieses Modell eignet sich am besten für eher kleinere Applikationen, die bereits produktiv genutzt werden und «fertig» sind. Auch für technisch orientierte Applikationen respektive Komponenten, die potenziell eine breitere, im vornherein nicht genau definierbare Zielgruppe ansprechen, ist dies meist die beste Form.

3.2 Einbindung der Lieferanten

Bei der Einbindung der Lieferanten gibt es grundsätzlich zwei Möglichkeiten:

a) *Lieferant als Auftragnehmer*: Er wird grundsätzlich als mittelfristig ersetzbar betrachtet. Der Lieferant soll die Applikation basierend auf Aufträgen des Kantons (oder der Community) kosteneffizient und in hoher Qualität weiterentwickeln, hat aber höchstens beratend Einfluss auf die Roadmap und die Priorisierung.

b) *Langfristige Partnerinnen und Partner mit Mitbestimmung*: Sie sollen bei der Entwicklung aktiv mitbestimmen können. Dadurch wird ihre Rolle gestärkt und sie sind potenziell bereit, auch selber in die Applikation zu investieren. Typischerweise vertreiben in diesem Modell die Lieferanten dann die Software und suchen aktiv nach neuer Kundschaft.

Nimmt der Lieferant eine starke Rolle ein, bringt dies der Kantonsverwaltung Vorteile: Durch die aktive Vermarktung der Applikation besteht die Chance, dass die Community schneller und stärker wächst und somit die Kosten pro Mitglied schneller sinken.

Wenn ein Modell mit einer starken Mitbestimmung der Lieferanten gewählt wird, muss darauf geachtet werden, dass dadurch keine Abweichung zu beschaffungsrechtlich vorgeschriebenen Regelungen entsteht. Es empfiehlt sich meist mindestens zwei Lieferanten mit einzubinden, damit keine zu starke Abhängigkeit zu einem einzelnen Lieferanten entsteht. Auch muss die Möglichkeit für einen Beitritt weiterer Lieferanten zur Community bestehen.

3.3 Verteilung der Kosten

Zur Aufteilung der Kosten für Wartung, Weiterentwicklung und neue Funktionen gibt es grundsätzlich zwei Varianten:

a) *Jeder seine*: Jeder bezahlt für die von ihm gewünschten Changes selber. Allenfalls wird auf einer Fall-zu-Fall-Basis entschieden, gewisse Erweiterungen gemeinsam zu bezahlen.

b) *Kostenteiler*: Die Kosten werden nach einem definierten Schlüssel aufgeteilt, grundsätzlich bezahlen alle für alles. Nur für ganz spezifische Erweiterungen wird davon abgewichen. Diese sollen direkt von der betreffenden Nutzerin oder vom betreffenden Nutzer bezahlt werden.
Der Kostenteiler kann zum Beispiel die Grösse des betreffenden Kantons oder die Anzahl der Nutzerinnen und Nutzer sein.

Grundsätzlich macht die Variante b (Kostenteiler) meist nur Sinn, wenn auch ein gemeinsames Produkt-Management wie bei Variante b (Gemeinsam) besteht. Nur wer mitbestimmen kann, wird auch bereit sein, die Folgekosten der Entscheide zu tragen.

4 Inhalte für das Community-Konzept

Dieses Kapitel gibt Hilfestellungen zum Schreiben des Community-Konzepts basierend auf den getroffenen Grundschatzentscheiden (gemäss Kapitel 3).

Produkt Management:
a) Kanton Bern
b) Gemeinsame Erarbeitung
c) Offene Organisation

Einbindung Lieferant:
a) Lieferant als Auftragnehmer
b) Lieferant als Partner

Verteilung der Kosten:
a) Jeder seine
b) Kostenteiler

Wo je nach getroffenen Grundsatzentscheiden unterschiedliche Inhalte sinnvoll sind, verwenden wir die folgende Tabellenstruktur. Der *kursive Text* ist als Vorlage zur direkten Übernahme in das Community-Konzepts gedacht.

PM[2]	L[3]	K[4]	Vorschlag
c	b	a	Inhalte oder Anregungen, wenn die Grundsatzentscheide lauten: Produkt Management: c) Offene Organisation Einbindung Lieferant: b) Lieferant als Partner Verteilung der Kosten: a) Jeder seine *Dieser Text kann in die Vorlage kopiert und angepasst werden.*
b	-	b	Inhalte oder Anregungen, wenn die Grundsatzentscheide lauten: Produkt Management: b) Gemeinsame Erarbeitung Einbindung Lieferant: a) oder b) (spielt keine Rolle) Verteilung der Kosten: b) Kostenteiler

Das Community-Konzept soll dabei ein dynamisches Dokument sein, das jederzeit angepasst werden kann. Es ist wenig sinnvoll, zu weit in die Zukunft zu schauen. Stattdessen muss das Konzept vor allem auf die aktuelle Situation angepasst sein. Wenn dann später mehr Mitglieder der Community beitreten, dann können die Organisation und Prozesse weiter ausgearbeitet und komplexere Mechanismen eingeführt werden.

Die Unterkapitel richten sich direkt nach der vorgeschlagenen Struktur des Community-Konzepts (siehe Kapitel 2).

[2] Produkt Management
[3] Einbindung Lieferanten
[4] Verteilung der Kosten

4.1 Zielsetzung

Die Zielsetzung soll einerseits enthalten, was der eigentliche Zweck bzw. die «Vision» der Applikation ist. Diese Vision soll auch in die Datei «README.md» im Repository einfliessen (vgl. «Checkliste Source Code und Dokumentation»).

Andererseits soll beschrieben werden, was aus Sicht der Community angestrebt wird:

- Wer soll beitreten? Wer sind mögliche Interessentinnen und Interessenten?
- Soll aktiv nach neuen Mitgliedern gesucht werden?
- Welche Hauptziele werden mit der Veröffentlichung als Open Source angestrebt?

4.2 Organisation

PM	L	E	Vorschlag
a	-	a	Da der Kanton Bern selber direkt die Kontrolle behält, ist kein Aufbau zusätzlicher Organisationen notwendig. Die publizierende Stelle (z. B. ein Amt) behält das Eigentum und übernimmt alle Koordinationsaufgaben. Das Kapitel «Organisation» kann in diesem Fall sehr kurz gehalten werden.
b	-	a	Wir empfehlen für diesen Fall meistens die Organisation als einfache Gesellschaft. Damit die Roadmap und die Anforderungen gemeinsam erarbeitet und abgestimmt werden können, empfiehlt sich die Schaffung zweier Gruppen, in denen pro Community-Mitglied je eine Teilnehmerin oder ein Teilnehmer dabei ist: • Management Board: Definiert die Strategie und die Priorisierung • Fachgruppe: Erarbeitet auf Expertenebene die Anforderungen und konkreten Inhalte. Je nach Umfang kann auch pro Thema eine eigene Fachgruppe eingesetzt werden. Eigentümerin oder Eigentümer des Codes ist die Stelle, die den Code publiziert hat. Wenn die Strukturen komplexer werden oder mehr als fünf Anwenderinnen oder Anwender bzw. weitere Mitglieder beteiligt sind, kann geprüft werden, ob sich die Community besser als Verein organisieren sollte. Vergleichen Sie hierzu die Ausführungen in der Variante direkt unterhalb.
b	-	b	Es empfiehlt sich eine Organisation als Verein. Entweder wird spezifisch für die Applikation ein eigener Verein gegründet oder aber die Applikation wird an einen bestehenden Verein übergeben. Ein Verein deshalb, weil sonst die Aufteilung der Kosten aufwändig wird und wahrscheinlich im Rahmen der Koordinationsaufgaben genug Arbeit anfällt, um eine nebenamtliche Geschäftsführerin oder einen Geschäftsführer für den Verein zu beschäftigen. Dem Verein werden dann die Rechte am Code übertragen, er übernimmt das Community-Management und die Bestellung bei den Lieferanten. Der Rechtsdienst des KAIO unterstützt sie gerne bei der Erarbeitung der

			entsprechenden Statuten.
b	a	a	(ergänzend zu «b-_-b») Hier sind die Lieferanten keine Mitglieder des Vereins, sondern nur Auftragnehmer. Die Geschäftsführerin oder der Geschäftsführer soll nicht von einem Lieferanten gestellt werden.
b	a	b	(ergänzend zu «b-_-b») Die Lieferanten sind Mitglieder des Vereins, der Geschäftsführer kann auch von einem Lieferanten gestellt werden.
c	-	a	Es braucht keine grosse Beschreibung einer Organisation, da diese ja bewusst offen gestaltet werden soll. Es muss aber geregelt werden, wer Anfragen von aussen entgegennimmt und diese bearbeitet (Zuständigkeit). Ebenfalls soll bestimmt werden, ob und inwiefern externe Personen mit eingebunden werden können: • Keine direkte Einbindung: Externe können nur Vorschläge und Beiträge (als Pull Request) einbringen. • Einbindung als Beitragende: Externe, die häufig und gute Beiträge leisten, werden direkt mit eingebunden. • Übergabe an Externe möglich: Falls nach der Veröffentlichung Externe mehr zur Weiterentwicklung beitragen als der Kanton Bern, dann kann die Applikation auch an diese übertragen werden. Die publizierende Stelle behält das Eigentum. Diese Variante entspricht der Organisation «klassischer» Open-Source-Projekte, siehe zum Beispiel https://opensource.guide/leadership-and-governance/ für weiterführende Informationen.
c	a	a	(ergänzend zu «c-_-a») Der Lieferant sollte nur rein technische Koordinationsaufgaben (Code-Review, Beurteilung) übernehmen.
c	b	a	(ergänzend zu «c-_-a») Der Lieferant kann die Koordinationsaufgaben übernehmen.

4.3 Roadmap und Change Prozess

PM	L	E	Vorschlag
a	a	-	*Der Kanton Bern definiert die Roadmap und entscheidet darüber, welche Changes umgesetzt werden. Es kommen dabei die Standardprozesse des Amtes XY zum Einsatz.*
a	b	-	*Der Kanton Bern definiert unter Einbezug der Firma XY die Roadmap. Er entscheidet, welche Changes umgesetzt werden. Es kommen dabei die Standardprozesse des Amtes XY zum Einsatz.*
b	-	a	Der Prozess zur Erarbeitung der Roadmap und zur Genehmigung von Changes muss von den Mitgliedern der Gesellschaft respektive des Vereins bestimmt werden. Je nach Mitglieder-Struktur (Anzahl, Grössenverhältnis, etc.) sind andere Prozesse sinnvoll. Die Prozesse müssen aber Massnahmen zur Konfliktlösung und Mehrheitsfindung beinhalten.
b	-	b	Analog «b-_-a» mit dem Zusatz, dass auch ein Verteilschlüssel für die Kosten festgelegt werden muss. Es soll auch berücksichtigt werden, wie mit der Kostenbeteiligung an von den betreffenden Mitgliedern nicht gewünschten Anpassungen umgegangen werden soll. Gerade bei Vereinen mit ungleich grossen Mitgliedern kann es zu Situationen kommen, in denen ein grosses Mitglied (z. B. der Kanton Bern) einen wesentlichen Teil der Kosten für eine Erweiterung trägt, ohne dass es einen Nutzen aus dieser zieht.
c	-	-	Beispiel, passt nicht für alle Situationen *Auf die Erstellung einer Roadmap wird verzichtet, die Applikation erfüllt derzeit die Bedürfnisse.* *Über Changes wird in einer offenen Diskussion entschieden und ein Konsens gesucht. Der Stichentscheid liegt beim Eigentümer des Codes (Kanton Bern).*

4.4 Entwicklungsprozess

PM	L	E	Vorschlag
a	-	-	*Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den vom Kanton Bern gewählten Lieferanten. Nach Vertragsablauf werden die betreffenden Rechte wieder entzogen.* *Die Lieferanten sind dafür zuständig, für externe Beiträge, die vom Kanton Bern fachlich akzeptiert wurden, den technischen Code-Review-Prozess durchzuführen. Sie haben die Verantwortung für die technische Qualität. Die Lieferanten werden für diese Arbeit vom Kanton Bern entschädigt.*
b	a	a	*Grundsätzlich erhalten alle von einem Mitglied der Community beauftragten Lieferanten Schreibzugriff auf den Code (Committer-Rechte). Nach Vertragsablauf werden die betreffenden Rechte wieder entzogen.* *Wo geänderter Code Kernbereiche der Applikation betrifft, müssen alle Änderungen von einem von der Community dafür speziell beauftragten Lieferanten reviewt werden. Dieser Lieferant hat die Verantwortung für die technische Qualität der Applikation. Er ist auch dafür zuständig, Änderungen zu reviewen, die von der Community fachlich akzeptiert wurden.*
b	a	b	*Die Committer-Rechte (Schreibzugriff auf den Code) liegen bei den Lieferanten, die Mitglied der Community sind. Die Lieferanten organisieren sich selber und tragen gemeinsam die Verantwortung für die Qualität des Codes.* *Wenn ein Community-Mitglied einen externen Lieferanten mit einer Anpassung oder Erweiterung beauftragt, dann erfolgt ein Review der Anpassung durch einen Lieferanten, der Mitglied der Community ist. Er wird dafür von der Auftraggeberin oder vom Auftraggeber entschädigt.* *Änderungen am Kern der Applikation müssen von einem zweiten Lieferanten, der Mitglied der Community ist, reviewt werden.* *Die Lieferanten, die Community-Mitglieder sind, sind ebenfalls für den Review von externen Beiträgen und neuem technischem Code zuständig, die vom Verein fachlich akzeptiert wurden.*
c	-	-	*Die Committer-Rechte liegen initial bei den Entwicklerinnen und Entwicklern des beauftragten Lieferanten. Es werden Committer-Rechte an alle Entwicklerinnen und Entwickler vergeben, die während mehreren Monaten aktiv zum Projekt beitragen und eine entsprechende Sozialkompetenz aufweisen.* *Die einzelnen Entwicklerinnen und Entwickler behalten grundsätzlich ihre Committer-Rechte, auch wenn sie ihren Arbeitgeber wechseln oder der Kanton Bern einen anderen Lieferanten wählt. Committer-Rechte verfallen nur, wenn sie freiwillig abgegeben werden oder die Mehrheit der anderen Committer den Entzug beschliesst.* *Der Kanton Bern hat das Recht, einzelne Entwickler der von ihm beauftrag-*

			ten Firmen als Committer zu bestimmen. Er hat nicht das Recht, jemandem Committer-Rechte grundlos zu entziehen. *Code-Reviews erfolgen durch mindestens eine Person, die als Committer tätig ist. Committer organisieren sich selber. Der Kanton Bern bekennt sich dazu, die Review-Arbeit zu entschädigen, soweit sie für ihn finanzierbar ist.*

4.5 Kostenteilung und Lieferanten

PM	L	E	Vorschlag
a	a	a	*Die Lieferanten werden periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) gewählt.*
a	b	-	Wichtig: Vorher prüfen ob dies im konkreten Fall beschaffungsrechtlich zulässig ist. *Der Kanton Bern wählt den Lieferanten XXX als strategischen Partner und strebt eine langfristige Zusammenarbeit an.*
b	a	a	*Jedes Mitglied der Community beauftragt in eigener Regie einen oder mehrere Software-Lieferanten zur Umsetzung der von ihm gewünschten Anpassungen und Erweiterungen.* *Für Anpassungen, die von mehreren Mitgliedern gewünschten werden, wird auf einer Fall-zu-Fall-Basis eine Kostenteilung vereinbart. Das Mitglied mit dem grössten Anteil der Kosten übernimmt die Auswahl und Beauftragung des Lieferanten.* *(Regelungen zur Wartung der Software)*
b	a	b	*Der Verein wählt die Software-Lieferanten periodisch mittels WTO-Ausschreibung oder freihändigem Verfahren (je nach Umfang der Wartungsleistungen) zentral.* *Die Kosten werden nach folgendem Schlüssel auf die Mitglieder umgelegt: (zu definieren: nach Bevölkerung, Benutzer etc.)*
b	b	a	Analog «b-a-a» allenfalls ergänzt um einen Beitrag der teilnehmenden Lieferanten. Beispiel: *Jede Software-Entwicklungsfirma, die Mitglied der Community ist, verpflichtet sich, pro Jahr mindestens XY Arbeitstage auf eigene Kosten in die Weiterentwicklung, technologische Aktualisierung oder Vermarktung der Applikation zu investieren.*
b	b	b	Analog «b-a-b», allenfalls ergänzt um den Zusatz gemäss «b-b-a».
c	a	-	Analog «a-a-a» mit dem Zusatz: *Die von den potenziellen Lieferanten beigetragenen Verbesserungen und*

			die Aktivitäten in der Community werden dabei als Faktor in der Bewertung berücksichtigt. *Von Externen gewünschte Changes werden nicht vom Kanton Bern finanziert, sondern müssen von diesen selber beauftragt und bezahlt werden.*
c	b	-	Analog «a-b-a» mit dem Zusatz: *Vom Lieferanten wird eine aktive Mitarbeit in der Community erwartet, auch wo diese Aufwände nicht direkt entschädigt werden.* *Von Externen gewünschte Changes werden nicht vom Kanton Bern finanziert, sondern müssen von diesen selber beauftragt und bezahlt werden. Der Lieferant erklärt sich bereit, diese auf Wunsch des Externen zu fairen Preisen umzusetzen.*

4.6 Vermarktung

PM	L	E	Vorschlag
a	a	-	*Wir unternehmen keine Aktivitäten zur Vermarktung der Applikation, veröffentlichen diese aber auf den Plattformen für Open-Source-Software. In den Fachgremien weisen wir darauf hin, dass wir die Applikation publiziert haben.* *Falls sich interessierte Personen bei uns melden, prüfen wir, ob eine gemeinsame Community gegründet werden soll, oder ob wir mit ihrer Version der Software weiterarbeiten.*
b	b	-	ergänzend zu oben. *Die Lieferanten dürfen die Applikation vermarkten und weitere Interessentinnen und Interessenten suchen. Der Kanton Bern darf als Referenz genannt werden. Wir nehmen bewusst in Kauf, dass dabei abweichende Versionen der Software entstehen.*
b	a	a	Die Applikation wird von den Mitgliedern oder dem Verein vermarktet.
b	b	a	Die Applikation wird von den Lieferanten vermarket.
b	-	b	Die Applikation wird vom Verein und dessen Mitgliedern vermarktet.
c	-	-	Die Applikation wird von den Lieferanten vermarket oder es wird auf eine explizite Vermarktung verzichtet. Variante ohne explizite Vermarktung: *Wir unternehmen keine Aktivitäten zur Vermarktung der Applikation, veröffentlichen diese aber auf den Plattformen für Open-Source-Software. In den Fachgremien weisen wir darauf hin, dass wir die Applikation publiziert haben und ihre Mitarbeit erwünscht ist.*

			Interessenten führen wir die Software vor und versuchen sie in unsere Prozesse zur Gestaltung der Roadmap miteinzubinden.

4.7 Umgang mit externen Beiträgen

PM	L	E	Vorschlag
a	-	-	Externe Beiträge und Anfragen müssen auch bearbeitet werden, wenn der Kanton der alleinige Entscheider über die Software bleibt. Wir schlagen folgenden Inhalt vor: *Umgang mit gemeldeten Fehlern* *Wir versuchen, von externen Personen gemeldete Fehler zu reproduzieren und fragen dabei falls nötig nach Präzisierungen. Falls die Reproduktion gelingt, dann beauftragt der Kanton Bern die Behebung der Fehler beim Lieferanten.* *Kann der Fehler nicht nachvollzogen werden oder ist der Aufwand zur Behebung unverhältnismässig hoch, dann entschuldigen wir uns beim Meldenden und fragen, ob ein Pull Request von ihm zur Behebung des Fehlers möglich wäre.* *Umgang mit Anfragen* *Wir reagieren auf jede eingehende Anfrage. Falls sich eine Anfrage nicht mit vernünftigem Aufwand klären lässt und die Aufwendung weiterer Zeit nicht effizient erscheint, dann entschuldigen wir uns mit dem Hinweis auf beschränkte Mittel und bieten der anfragenden Person an, mit einem Lieferanten Kontakt aufnehmen zu können für eine weitere Beratung.* *Umgang mit Pull Requests* *Wir prüfen jeden Pull Request wohlwollend. Um unnötigen Aufwand seitens des Beitragenden zu vermeiden, weisen wir gleich zu Beginn darauf hin, wenn etwas unserer Roadmap widerspricht oder es fraglich ist, dass wir den Beitrag akzeptieren werden. Der Kanton Bern entscheidet alleine und nach fachlichen Kriterien, welche Beiträge akzeptiert werden. Immer akzeptiert werden Fehlerkorrekturen, die den Review-Prozess bestanden haben.* *Umgang mit Forks* *Wir unterstützen Forks von unserer Applikation. Wer die Applikation einsetzt, soll selber einen Fork davon erstellen. Wir schauen mindestens einmal pro Jahr die entstandenen Forks an und übernehmen gute und für uns passende Erweiterungen.*
b	-	-	Extern bezieht sich hier auf Beiträge von ausserhalb der definierten Com-

			munity (Mitglieder im Verein). Abweichungen zu oben: *Umgang mit Forks* *Wir streben an, dass keine (langlebigen) Forks der Applikation entstehen. Stattdessen sollen Interessierte der Community direkt beitreten. Innerhalb der Community versuchen wir, dass alle Mitglieder die Hauptversion verwenden.*
c	-	-	*Für uns ist der Aufbau einer funktionierenden Community und einer offenen Kultur wichtig.* *Umgang mit gemeldeten Fehlern* *Wir versuchen gemeldete Fehler zu reproduzieren und zu korrigieren. Hierbei fordern wir die aktive Mitarbeit der meldenden Personen ein. Falls es für sie möglich ist, sollen sie direkt einen Pull Request mit einer Fehlerkorrektur erstellen.* *Umgang mit Anfragen* *Wir behandeln jede Anfrage innert maximal einer Woche. Anfragen von aktiv Beitragenden behandeln wir prioritär und vertieft. Wir versuchen, andere Beitretende in die Behandlung der Anfragen miteinzubeziehen.* *Umgang mit Pull Requests* *Wir versuchen möglichst viele und gute Pull Requests zu bekommen. In die Reviews und technische sowie fachliche Beurteilung der Pull Requests binden wir alle aktiv Beitragenden ein. Bei kontroversen Beiträgen versuchen wir mit informellen Abstimmungen (Votings) einen Entscheid zu finden.* *Umgang mit Forks* *Wir versuchen, dass Forks möglichst nicht nötig sind, da wir ein aktives und offenes Community-Management betreiben. Falls Forks entstehen, schauen wir diese aktiv an und versuchen die da entstandenen Erweiterungen und Verbesserungen zurückzugeben. Wir fragen in diesem Fall aktiv nach Pull Requests.* Weitere Anregungen für ein gutes Community-Management: https://opensource.guide/code-of-conduct/ und https://opensource.guide/best-practices/

4.8 Betrieb der Applikation

PM	L	E	Vorschlag
b	-	-	Der Betrieb kann zentral optional zusätzlich durch den zentralen Verein angeboten werden, im Sinne von Software-as-a-Service. Es soll geprüft werden, ob dies gewünscht wird.
-	b	-	Festhalten, ob der Lieferant die Software auch gleich betreiben soll

5 Verzeichnis von bestehenden Community-Konzepten

Im Sinne einer Sammlung von Best-Practices sind nachfolgend bereits erarbeitete Community-Konzepte sowie vergleichbare Dokumente aufgeführt:

- GERES-Community (http://geres-community.ch)
  Einordnung:
  - Produkt-Management: a) Gemeinsame Erarbeitung
  - Lieferant: eher a) Auftragnehmer
  - Verteilung der Kosten: b) Kostenteiler

  Anmerkung: Das Gemeinderegister GERES ist nicht Open Source, jedoch können viele Aspekte aus der Organisation auch für Open-Source-Applikationen relevant sein. Dokumente: Statuten (öffentlich) und Geschäftsordnung (auf Anfrage).

- iGov Portal (https://www.igovportal.ch)

6 Dokument – Protokoll

Dokumentnummer 274637

Autor Joos Thomas, FIN-KAIO-AP-SW

Änderungskontrolle

Version	Name	Datum	Bemerkungen
1.5	Mario Siegenthaler	14.05.2018	Erste Fassung
0.2	Thomas Joos	22.05.2018	Überarbeitung
0.3-0.4	Mario Siegenthaler	08.06.2018	Überarbeitung
0.5	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	22.03.2023	Aktualisierung
1.5.1	Nicolas Steiner	14.04.2023	Aktualisierung

Prüfung

Version	Stelle	Datum	Visum	Bemerkung
0.5	Thomas Joos	13.06.2018	Tjo	---
0.6	PB	02.07.2018	PB	Genehmigung Portfolioboard
1.2	Stab	13.09.2018	ssc	Sprachliche Prüfung

Freigabe

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