

# Der Kitodo Entwicklungsfonds

## Open-Source-Softwareentwicklung kooperativ finanzieren und gestalten

Matthias Finck, Hochschule für Angewandte Wissenschaften Hamburg

Robert Strötgen, Technische Universität Braunschweig

### Zusammenfassung

Open-Source-Software ist von Alterungsprozessen bedroht, die ohne eine bewusste Steuerung die Nachhaltigkeit der Software gefährden. Dieser Beitrag stellt den Kitodo-Entwicklungsfonds als Lösungsansatz vor, diesem Alterungsprozess kontinuierlich zu begegnen. Kitodo ist dreierlei – eine verbreitete Open-Source-Lösung zur Produktion und Präsentation digitaler Objekte, eine Community und ein gemeinnütziger Verein. Kitodo.Production ist das Workflowmanagementmodul der Kitodo-Suite, dass in Deutschland in mehr als 50 Institutionen eingesetzt wird. Der Entwicklungsfonds stellt ein Modell dar, wie Verein und Community gemeinsam erfolgreich eine Finanzierung und Steuerungsprozesse zur Lösung zur Vermeidung der Überalterungsgefahr etabliert haben.

### Summary

Open source software is threatened by aging processes that endanger the sustainability of the software without targeted control. This paper presents the Kitodo Development Fund as an approach to address this aging process in a continuous way. Kitodo is three things in one – a widely used open source solution for producing and presenting digital objects, a community, and a non-profit association. Kitodo.Production is the workflow management module of the Kitodo suite that is used in more than 50 institutions in Germany. The development fund represents a model of how the association and the community together have successfully established a funding and controlling process to solve the risk of obsolescence.

**Zitierfähiger Link (DOI):** <https://doi.org/10.5282/o-bib/5845>

**Autorenidentifikation:** Strötgen, Robert: ORCID: <https://orcid.org/0000-0003-3320-5187>;

Finck, Matthias: ORCID: <https://orcid.org/0000-0002-0848-2935>

**Schlagwörter:** Open Source Software ; Nachhaltige Softwareentwicklung ; Kitodo ; Digitalisierung ; Entwicklungsfonds

Dieses Werk steht unter der Lizenz [Creative Commons Namensnennung 4.0 International](https://creativecommons.org/licenses/by/4.0/).

# 1. Einleitung

Kitodo.Production ist das Workflowmanagementmodul der Kitodo-Suite<sup>1</sup>. Es unterstützt den Digitalisierungsprozess von verschiedenen Materialarten wie Akten, Drucken, Periodika, Handschriften, Noten und Musikalien, Einblattmedien und Dokumentennachlässen und kann Workflows hoch individualisiert abbilden. Kitodo.Presentation ist das zugehörige Modul für die Zugänglichmachung von digitalisierten Objekten in einem Webportal, aber auch über standardisierte Schnittstellen wie OAI-PMH<sup>2</sup> und IIIF<sup>3</sup>.

Ursprünglich ca. 2004 unter dem Namen Goobi aus einem Drittmittelprojekt hervorgegangen, wurde die Software bald in einer großen Zahl von Bibliotheken und anderen kulturbewahrenden Einrichtungen eingesetzt. Zur Sicherung der Nachhaltigkeit und Offenheit nicht nur der Lizenz, sondern auch des Entwicklungsprozesses, wurde 2012 der Verein Goobi. Digitalisieren im Verein e.V. gegründet, der 2016 angesichts des Scheiterns des Bemühens um eine einvernehmliche Lösung mit einem Softwaredienstleister in Kitodo. Key to digital objects e.V. umbenannt wurde.<sup>4</sup> Dieser Verein umfasst heute knapp 50 Mitglieder aus dem Bereich der Kultur- und Gedenkstätten sowie Entwicklungsdienstleistern<sup>5</sup>.

In den ersten Jahren der Vereinsarbeit galt der Schwerpunkt der Sicherung des offenen Entwicklungsprozesses. Dafür leistet sich der Verein ein von den Mitgliedern gewähltes Release-Management, das im Sinne des Vereins-Mantras der Offenheit eine faire Beteiligung der Mitglieder und der Community am Entwicklungsprozess und die Hoheit des Vereins über den Quelltext sichert.

Dieses Releasemanagement überhaupt finanzieren zu können (mit einem für die Releasemanager nicht kostendeckenden Beitrag) war bei Vereinsgründung nicht sicher. Und auch nach 10 Jahren und deutlichen Mitgliederzuwächsen lassen die Mitgliedsbeiträge neben dieser zentralen Vereinsaufgabe nur noch wenig Spielräume für weitere Vereinstätigkeiten wie Weiterbildung/Wissenstransfer und Förderung der Digitalisierung. Für aus Vereinsmitteln finanzierten Entwicklungsvorhaben reichten die Ressourcen nicht annähernd aus – und standen bei der Vereinsgründung auch nicht auf der Agenda. Für eine Modernisierung eines der zentralen Module (Kitodo.Production) war der Verein daher auf die Einwerbung von Drittmitteln angewiesen. Im Rahmen eines dreijährigen DFG-Projekts konnte die Software Kitodo.Production von 2016 bis Ende 2019 dann grundlegend überarbeitet werden.<sup>6</sup>

---

1 Vgl. Kitodo, 2022. <<https://www.kitodo.org>>, Stand: 08.08.2022.

2 Das „Open Archives Initiative Protocol for Metadata Harvesting“ (OAI-PMH) ist ein Protokoll zum Harvesten von Metadaten, siehe: <<https://www.openarchives.org/pmh/>>, Stand: 08.08.2022.

3 Das „International Image Interoperability Framework“ (IIIF) ist ein Rahmenwerk zum institutionsübergreifenden Austausch digitaler Objekte sowie ihrer standortunabhängigen Darstellung in unterschiedlichsten Präsentationslösungen, siehe <<https://iiif.io/>>, Stand: 08.08.2022.

4 Vgl. Bonte, Achim: Chancen und Fallstricke offener, kooperativer Softwareentwicklung. Das Beispiel Kitodo, in: Bonte, Achim; Rehnolt, Juliane (Hg.): Kooperative Informationsinfrastrukturen als Chance und Herausforderung, Berlin 2018, S. 182–192. Online: <<https://doi.org/10.1515/9783110587524-022>>, Stand: 08.08.2022.

5 Vgl. Kitodo, Mitglieder, 2022. <<https://www.kitodo.org/verein/mitglieder>>, Stand: 08.08.2022.

6 Vgl. Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019. Online: <<https://doi.org/10.5281/zenodo.6779479>>.

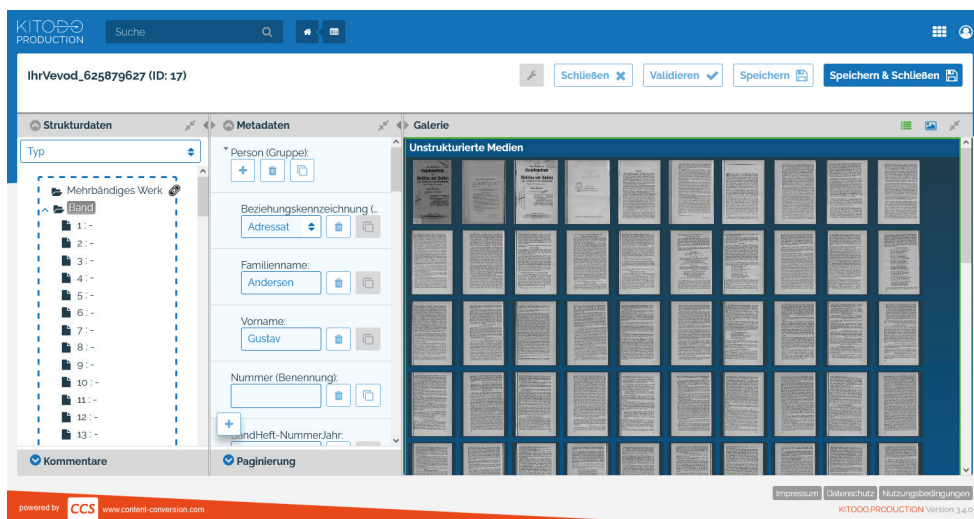


Abb. 1: Screenshot der grundlegend überarbeiteten Version von Kitodo.Production

Dabei musste der gesamte technische Kern der Software und die grundlegende Architektur neu konzipiert und aufgebaut werden, da die vorherige Version technisch so veraltet war, dass eine schrittweise Modernisierung nicht mehr möglich erschien. Die Software wurde mit einem hohen personellen und finanziellen Aufwand auf der Basis einer modernen, modularen Architektur neu aufgebaut.<sup>7</sup>

2019 startete eine Diskussion in der Kitodo-Community, wie sich die hohen Entwicklungskosten in Bezug auf Kitodo.Production 3.0 für die Zukunft vermeiden lassen könnten. Nach dem erfolgreichen Ende des Relaunch-Projektes galt es zu klären, wie sich für die nun neue und technisch moderne Version die Wiederholung eines schleichenden Alterungsprozesses verhindern und die Software kontinuierlich modernisieren lässt.

Im Folgenden werden wir die Ursachen für diese oftmals unbehandelten Alterungsprozesse bei Open Source Software aufzeigen und mit dem Entwicklungsfonds der Kitodo Community einen Lösungsansatz präsentieren, der diese Ursachen adressiert.

## 2. Der Alterungsprozess von Open Source Software

Im Gegensatz zu proprietärer Software wird Open Source Software zumeist nicht durch ein zentrales Unternehmen, sondern durch eine verteilte Community getragen. Diese Community ist u.a. durch Verteiltheit und Offenheit geprägt.<sup>8</sup> Während in einem zentral gesteuerten proprietären

7 Vgl. Meyer, Sebastian; Huber, Kathrin: Konsequent Modular – ein offenes, modernes Architekturkonzept, in: Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019, S. 40–49. Online: <<https://doi.org/10.5281/zenodo.6779479>>.

8 Vgl. Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019. Online: <<https://doi.org/10.5281/zenodo.6779479>>.

Softwareentwicklungsprozess das Unternehmen die Entwicklungsprioritäten setzen kann, erfolgt dies in Open-Source-Projekten aus der verteilten Community heraus. Diese Form des Entwicklungsprozesses entspricht somit zumeist dem Issue-basierten Modell, in dem eine Menge von Problemfelder (Issues) durch die Community formuliert werden, die dann bearbeitet werden.<sup>9</sup> Die Issues kommen dabei aus der Anwender-Community heraus und haben durch ihre Herkunft immer eher die Funktionsvielfalt und die Benutzungsschnittstelle im Blick, als die Architektur oder den technischen Kern der Software. So wählen die Entwickler\*innen von Open-Source-Projekten aus den Issues dann diejenigen aus, die zu ihren persönlichen Interessen und Fähigkeiten passen<sup>10</sup>, weil intrinsische Motivation ein wesentlicher Faktor in der Open-Source-Entwicklung ist.<sup>11</sup>

Die Entwicklung von Open Source Software lässt sich somit am ehesten dem Feature Driven Development (FDD) Ansatz vergleichen.<sup>12</sup> FDD stellt den Feature-Begriff in den Mittelpunkt der Entwicklung. Jedes Feature stellt einen Mehrwert für die Anwender\*innen / Kund\*innen dar. Die Entwicklung wird anhand eines Feature-Plans organisiert. Eine wichtige Rolle im FDD spielt der oder die Chefarchitekt\*in<sup>13</sup>, die ständig den Überblick über die Gesamtarchitektur und die fachlichen Kernmodelle behält. Im Open-Source-Entwicklungsprozess kommt diese Rolle dem Releasemanagement zu. Anders als der oder die Chefarchitekt\*in ist das Releasemanagement allerdings nicht weisungsbefugt oder kann extrinsische Motivationsanreize – z.B. durch Bezahlung – setzen, sondern kann nur über die Coding-Guidelines und das Code-Review die Entwicklungsbeiträge steuern – bis hin zu Ablehnung eines Code-Beitrags. Es hat kaum Einfluss auf die Entwicklungsprioritäten.

Das führt oftmals dazu, dass der softwaretechnische Kern einer Open Source Software droht mit der Zeit zu veralten, während außen herum immer weiter neue Features entwickelt werden. Das Refactoring (die Überarbeitung) von Open Source Code findet eher selten statt.<sup>14</sup> Die Mittel für die Softwareentwicklung stammen i.d.R. von den Anwendenden und bei denen liegt der Fokus auf der Funktionserweiterung. So altert der softwaretechnische Kern i.d.R. so lange, bis die Wartbarkeit und die Sicherheitsrisiken ein Maß erreicht haben, die die Entwicklungskosten für funktionale Weiterentwicklung so steigen lassen, dass ein grundlegendes Refactoring unumgänglich ist. Diese dann notwendigen grundlegenden Überarbeitungsschritte sind aber mit so viel finanziellem oder

9 Vgl. Brügge, Bernd; Harhoff, Dietmar; Picot, Arnold u. a.: Open-Source-Software. Eine ökonomische und technische Analyse, Berlin 2004 (Springer eBook Collection Business and Economics). Online: <<https://doi.org/10.1007/978-3-642-17024-9>>.

10 Vgl. Hippel, Eric von; Krogh, Georg von: Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science, in: Organization Science 14 (2), 2003, S. 209–223. Online: <<https://doi.org/10.1287/orsc.14.2.209.14992>>.

11 Vgl. Hars, Alexander, & Ou, Shaosong (2002). Working for Free? Motivations for Participating in Open-Source Projects, in: International Journal of Electronic Commerce, 6(3) 2002, 25–39. Online: <<http://www.jstor.org/stable/27751021>> Stand: 08.08.2022.

12 Vgl. Coad, Peter; Lefebvre, Eric; DeLuca, Jeff: Java modeling in color with UML. Enterprise components and process; [enrich the content of your models, become a better modeler by example, deliver frequent, tangible, working results, Upper Saddle River, NJ 1999.

13 Vgl. Wolf, Henning; Bleek, Wolf-Gideon: Agile Softwareentwicklung. Werte, Konzepte und Methoden, Heidelberg 2011.

14 Vgl. Vassallo, Carmine; Grano, Giovanni; Palomba, Fabio u. a.: A large-scale empirical exploration on refactoring activities in open source software projects, in: Science of Computer Programming 180, 2019, S. 1–15. Online: <<https://doi.org/10.1016/j.scico.2019.05.002>>.

personellem Aufwand verbunden, dass sie ein Risiko des Scheiterns des gesamten Open-Source-Projekts in sich bergen.

In bibliothekarischen Open-Source-Projekten waren diese kritischen Entwicklungsphasen, in denen eine veraltete Softwareversion grundlegend überarbeitet werden musste, oftmals zu beobachten. Sie gingen mit erheblichen Verzögerungsprozessen in der Auslieferung der neuen Softwareversion einher und standen hinsichtlich ihres Erfolges oftmals auf der Kippe. Als Beispiele mögen hier die Entwicklung von DSpace<sup>15</sup>, VuFind<sup>16</sup> oder auch Kitodo.Production genannt sein.

## 3. Der Weg zum Kitodo Entwicklungsfonds

### 3.1 Ziele und Herausforderungen

In der Kitodo Community wurden aus diesem langjährigen, mühsamen und finanziell aufwändigen Relaunchprozess von der Version Kitodo.Production 2 zur Version Kitodo.Production 3 die Lehren gezogen. Entsprechend wurden Maßnahmen ergriffen, in Zukunft trotz einer Anlehnung an Feature Driven Development (FDD) und einem möglichst offenen Entwicklungsprozess, den Alterungsprozess des softwaretechnischen Kerns aktiv zu begleiten und kontinuierlich zu modernisieren. Als Lösungs-idee schlugen die Beteiligten bereits zum Projektende den Aufbau eines Entwicklungsfonds vor.<sup>17</sup>

Der zentrale Gedanke hinter dem Entwicklungsfonds ist es, Geld für die Umsetzung von Issues zur Verfügung zu stellen, die nicht primär den Entwicklungswünschen der Anwender nach Funktionsverbesserung oder einer Erhöhung der Usability gerecht werden, sondern durch die Bereitstellung der finanziellen Mittel einen Anreiz für die Bearbeitung von Aufgaben im Bereich der Pflege des software-technischen Kerns zu schaffen.

Während die Umsetzung von neuen Features klassischerweise anlassbezogen in abgeschlossenen Projekten stattfindet, könnte durch einen Entwicklungsfonds eine dauerhafte Finanzierung für die kontinuierlichen Wartungs- und Pflegeprozess der Software sichergestellt und so die Wiederholung der Situation von 2016 vermieden werden. Der Entwicklungsfonds von Kitodo soll also der Umsetzung von Aufgaben, wie den folgenden dienen:

- Umstellung auf neue Software-Versionen (TYPO3, Java, u.v.m.)
- Performance-Verbesserungen
- Behebung kritischer Bugs
- Entwicklung zentraler Funktionen oder Schnittstellen

15 Vgl. DSpace, The Software of Choice for Academic, Non-Profit & Commercial Organizations Building Open Digital Repositories. <<https://dspace.lyrasis.org/>>, Stand: 08.08.2022.

16 Vgl. VuFind, Release 8.1, 18.07.2022. <<https://vufind.org/vufind/>>, Stand: 08.08.2022.

17 Vgl. Huber, Kathrin; Strötgen, Robert: Und nun? Der Weg vom Projekt zur kontinuierlichen Produktentwicklung, in: Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019, S. 72–80. Online: <<https://doi.org/10.5281/zenodo.6779479>>.

Auf dem Weg zu dem Entwicklungsfonds ergaben sich zwei zentrale Herausforderungen:

Zum einen musste die Offenheit, Transparenz und vor allem der Partizipationsgedanke der Open Source Community bewahrt bleiben. Der Fonds muss als gemeinsame Finanzierung aus der Community heraus etabliert werden, bei der allen Mitgliedern klar ist, für welche Issues die Mittel verwendet werden. Und es muss einen demokratischen Prozess geben, die Mitglieder an dem Auswahlprozess zu beteiligen.

Zum anderen mussten die Mitglieder der Community motiviert werden, sich an den Kosten des Fonds zu beteiligen. Als notwendige Größenordnung wurde mit einem Volumen von ca. 70.000 € pro Jahr kalkuliert, die notwendig sein würden, um den Anforderungen an eine kontinuierliche Pflege gerecht zu werden. Diese Schätzung des Volumens ergab sich aus den Erfahrungswerten jährlicher Entwicklungs- und Pflegebudgets anderer Projekte in den beteiligten Institutionen, die einen zeitlichen Aufwand von mindestens 50 bis 60 Std. pro Monat nahelegten. Mit einem bei Dienstleistern üblichen Stundensatz von 100,00 € pro Stunde ließ sich die entsprechende Summe von ca. 70.000 € ermitteln. Um der gesamten Softwaresuite von Kitodo gerecht zu werden, wurde das Präsentationmodul Kitodo.Presentation mit in die Planungen für einen Entwicklungsfonds aufgenommen.

### 3.2 Ein erstes Modell

Um diesen beiden Herausforderungen gerecht zu werden, entwickelte der Vorstand des Kitodo e.V. 2019 folgendes Finanzierungs- und Prozessmodell:

Entwicklungsfonds Kitodo.Production		Entwicklungsfonds Kitodo.Presentation	
Höhe des Fonds:	mind. 50.000€	Höhe des Fonds:	mind. 20.000€
Förderbeitrag pro Stimme:	2.500€	Förderbeitrag pro Stimme:	1.000€
Juniormitgliedschaft:	2.500 €/1 Stimme	Juniormitgliedschaft:	1.000 €/1 Stimme
Standardmitgliedschaft:	5.000 €/2 Stimmen	Standardmitgliedschaft:	2.000 €/2 Stimmen
Premiummitgliedschaft:	10.000 €/4 Stimmen	Premiummitgliedschaft:	4.000 €/4 Stimmen

Abb. 2: Ein erstes Finanzierungsmodell durch gestaffelte Mitgliedschaften

Die erste Version des Finanzierungsmodells sah vor, dass die Mitglieder von Kitodo durch einen gestaffelten Beitrag auch Mitglieder des Entwicklungsfonds werden konnten. Ein möglicher Förderbeitrag wurde auf 1.000 € (Juniormitgliedschaft bei Kitodo.Presentation) bis 10.000 € (Premiummitgliedschaft bei Kitodo.Production) festgesetzt.

Dabei wurde für die beiden Module Kitodo.Production und Kitodo.Presentation jeweils mit einem eigenständigen Fonds geplant, da viele Mitglieder in der Community nur eines der Module nutzen. Außerdem wurde die Vergabe von Stimmen an die Höhe des Betrags gekoppelt, sodass ein höherer finanzieller Beitrag ein bis zu vierfaches Stimmrecht gegenüber einem niedrigen Beitrag erhielt.

Um die Verteilung der so eingesammelten Mittel transparent und partizipativ zu gestalten, wurde ein erster Vorschlag eines sich jährlich wiederholenden vierstufigen Prozesses entwickelt:



Abb. 3: Prozessvorschlag für die Verwendung der Mittel

In der *ersten Stufe* sollte allen an Kitodo Interessierten die Möglichkeit eingeräumt werden, Vorschläge und Ideen einzureichen. Diese Möglichkeit beschränkte sich nicht nur auf Mitglieder des Vereins, sondern stand allen offen, indem auf GitHub18 – einem frei verfügbaren Dienst zur Versionsverwaltung für Software-Entwicklungsprojekte – die Wünsche als sogenannte Issues eingetragen und kommentiert werden konnten. So entstand eine Liste an potenziellen Aufgaben, die mit Hilfe des Entwicklungsfonds umgesetzt werden sollten.

In der *zweiten Stufe* hat das Releasemanagement (RM) eine Bewertung und sehr grobe Aufwandschätzung der Issues auf dieser Liste vorgenommen. Die Aufwandsabschätzung war geplant, damit die Mitglieder des Entwicklungsfonds bei der Priorisierung den zu erwartenden Aufwand bei ihrer Abstimmung berücksichtigen konnten. Die grundsätzliche Bewertung sollte dem Releasemanagement die Möglichkeit geben einzuschätzen, ob die Entwicklungswünsche mit den Zielen des Entwicklungsfonds übereinstimmen bzw. der Wunsch vielleicht schon umgesetzt wurde oder nicht umsetzbar ist.

Die *dritte Stufe* war als der eigentliche Abstimmungsprozess der Mitglieder aus dem Entwicklungsfonds (EF) geplant, die an einem gemeinsamen Termin noch einmal zusammen über die Liste potenzieller Issues diskutieren konnten, um dann mit dem jeweiligen Stimmgewicht, das sich aus der Höhe des Beitrags ergab, abzustimmen. Aus der Anzahl der Stimmen für ein Issue ergibt sich ein Ranking von mehr oder weniger hoch priorisierten Entwicklungswünschen.

Die *vierte Stufe* sah vor, dass die Geschäftsstelle (GS) des Vereins gemeinsam mit dem Releasemanagement Ausschreibungen für die höchst priorisierten Entwicklungswünsche erstellt und angelehnt an die Vergabeverfahren des öffentlichen Dienstes an Dienstleister vergibt. Die Anzahl der umsetzbaren Entwicklungswünsche ergibt sich dabei zum einen aus der priorisierten Liste mit den Aufwandsabschätzungen und zum anderen aus dem verfügbaren Budget des Entwicklungsfonds. Wünsche, die nicht im Rahmen des Budgets umgesetzt werden können, wandern wieder auf die Liste der Entwicklungswünsche für den nächsten Durchlauf.

18 Vgl. Kitodo. Key to digital objects, <<https://github.com/Kitodo>> Stand: 08.08.2022.

### 3.3 Evaluation und Überarbeitung des Konzepts

Das Prozessmodell und das Finanzierungskonzept wurden nach dem Entwurf innerhalb des Vereins und der Community diskutiert und zeitgleich rechtlich geprüft. Die Diskussion in der Community ergab, dass die ausdifferenzierte Stimmverteilung anhand der Beitragshöhe als Motivation zur Zahlung höherer Beiträge nicht notwendig war. Die große Mehrheit der Vereinsmitglieder – auch die, die bereits einen hohen Beitrag zugesagt hatten – hielt ein stark vereinfachtes Modell, in dem jedes Communitymitglied unabhängig vom Mitgliedsstatus oder der Beitragshöhe eine Stimme hat, für angemessen und ausreichend. Hier konnte das Modell also deutlich verschlankt werden.

Gleichzeitig zeigte eine vereinsrechtliche Prüfung erhebliche steuerliche Herausforderungen und außerdem eine Gefährdung der Gemeinnützigkeit des Vereins für den Fall, dass Mitglieder durch ihre Beiträge fest mit dem Stimmrecht für die Priorisierung des Entwicklungsfonds (Kitodo.Production oder Kitodo.Presentation) direkte geldwerte Vorteile erhalten könnten. Auch hier führte die endgültige Lösung zu einer deutlichen Vereinfachung des ursprünglichen Modells. In Zukunft können die Vereinsmitglieder gestaffelte Mitgliedsbeiträge für den Verein bezahlen. Diese Vereinsbeiträge sind nicht zweckgebunden, werden aber vom Vorstand dem Entwicklungsfonds zugeführt.

Ansonsten stieß das vorgeschlagene Modell auf breite Zustimmung, sodass das finale Prozess- und Beitragsmodell nur eine Vereinfachung des ersten Entwurfs darstellte:

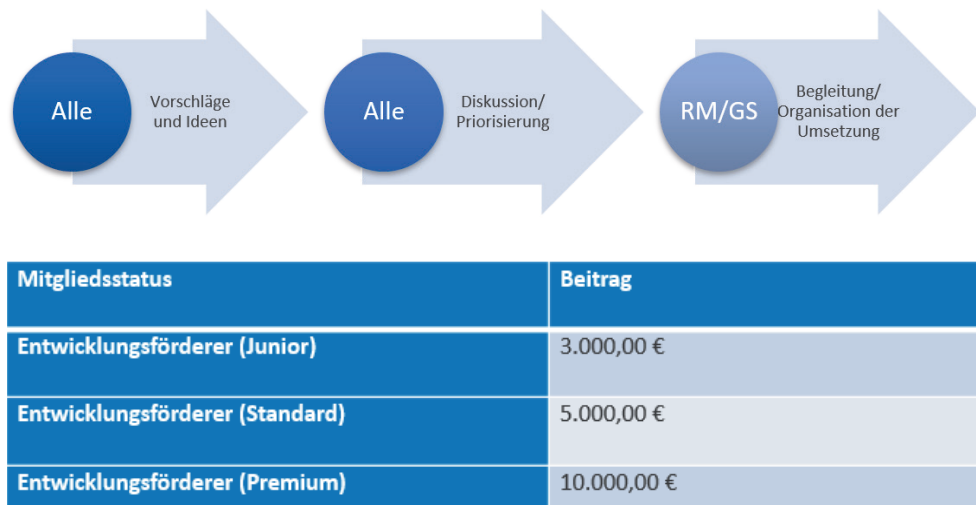


Abb. 4: Das finale Prozess- und Beitragsmodell für den Entwicklungsfonds

Aus den vier vorgesehenen Prozessschritten wurden als Konsequenz aus der Evaluation drei. Eine erste Stufe zur Sammlung von Vorschlägen und Ideen, eine zweite Stufe, bei der alle gemeinsam die vom Releasemanagement nur noch zeitlich abgeschätzten Ideen diskutieren und priorisieren



und eine dritte Stufe für die Umsetzung. Auf die Differenzierung zwischen Kitodo.Production und Kitodo.Presentation wurde verzichtet, was auch den Vorteil höherer Flexibilität mit sich bringt. Aus dem nach den zwei Modulen und der Stimmenzahl differenzierten Finanzierungskonzept, wurde so ein einfach gestaffeltes System an Mitgliedsbeiträgen ohne Einfluss auf die Anzahl der Stimmen.

## 4. Der erste Durchgang des Entwicklungsfonds

Mit Beschluss der Mitgliederversammlung und Anpassung der Beitragsordnung im November 2020 hat der Verein die Grundlagen dafür gelegt, dass der Entwicklungsfonds seine Arbeit aufnehmen und seinen Zweck erfüllen kann. Zehn Mitglieder hatten bis zum Jahresbeginn 2021 (zusätzliche) Mitgliedsbeiträge in Höhe von 47.000 € zugesagt, die vollständig dem Entwicklungsfonds zu Gute kommen sollten.

### 4.1 Schritt 1: Sammlung der Ideen

Im Februar 2020 erfolgte der Aufruf zur Benennung von Vorschlägen über die Mailinglisten des Vereins und der Community. Die Vorschläge wurden auf GitHub gesammelt (von den Vorschlagenden selbst oder mit Unterstützung des Releasemanagements), über entsprechende Labels getaggt und dort auch direkt von Entwickler\*innen und Community kommentiert und diskutiert. Zur Einschätzung des Aufwands (und damit auch der zu erwartenden Kosten) wurden die Vorhaben nach einem einfachen Schema kategorisiert. Dabei kamen 6 Vorschläge für Kitodo.Presentation und 13 Vorschläge für Kitodo.Production zusammen.

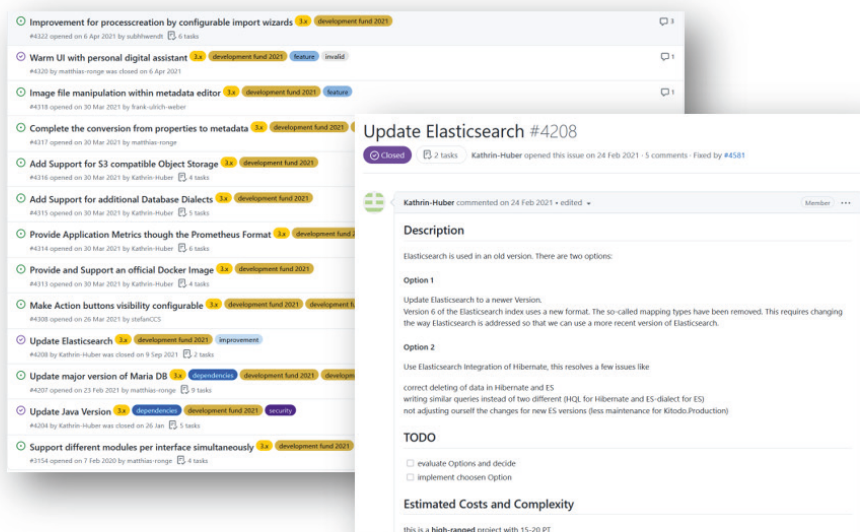


Abb. 5: Die Entwicklungsvorschläge für Kitodo.Production und eine exemplarische Kostenabschätzung

## 4.2 Schritt 2: Gemeinsame Diskussion und Priorisierung

Nach zwei Monaten der Sammlung und Diskussion fand im April 2022 unter Leitung des Vereinsvorstands und Beteiligung des Releasemanagements als Webkonferenz ein Treffen zur Diskussion und Priorisierung der Vorschläge statt, zu dem offen über Verein und Community eingeladen worden war. Jeder einzelne Vorschlag wurde vom Vorschlagenden vorgestellt und anschließend gemeinsam diskutiert.

Abschließend wurde mit Nutzung des Online-Votingtools PollUnit eine Priorisierung der Vorschläge durchgeführt, bei der alle Teilnehmenden insgesamt 7 Stimmen, pro Vorschlag maximal 3 Stimmen, verteilen konnte. Die Abstimmung funktionierte problemlos und führte zu einem klaren Ergebnis. Das Ranking priorisierte sowohl für Kitodo.Production als auch für Kitodo.Presentation vor allem Vorhaben, die eine Aktualisierung der genutzten Basis-Software zum Ziel hatten (für Kitodo.Production ein Update auf aktuelle Versionen von Java und Elasticsearch, für Kitodo.Presentation ein Update auf die aktuelle TYPO3-Version und ein Umstieg auf die entsprechende Template-Engine). Mit niedrigerer Priorisierung folgte ein weiteres Vorhaben zur Verbesserung der Funktionalität beim Anlegen neuer Vorgänge in Kitodo.Production im Ranking. In der abschließenden Diskussion als Retrospektive für den ersten Durchlauf der gemeinsamen Priorisierung wurde das Verfahren als sehr gut durchführbar und nützlich bewertet und es herrschte eine große Zufriedenheit mit dem Abstimmungsergebnis.

## 4.3 Schritt 3: Organisation und Umsetzung

Auf Grundlage der Priorisierung erstellte das Releasemanagement Leistungsbeschreibungen der einzelnen hoch priorisierten Vorhaben, mit denen dann der Vorstand eine offene Ausschreibung durchführte, die den Rahmenbedingungen öffentlicher Auftragsvergabe entspricht, auch wenn dies für den Verein nicht zwingend erforderlich gewesen wäre. Nach Eingang der Angebote und einer Entscheidung durch den Vereinsvorstand mit Beratung durch das Releasemanagement konnten die ersten drei Aufträge vergeben werden. Eine weitere Ausschreibung eines vierten Vorhabens konnte im Anschluss durchgeführt werden, da die durch die Aufträge gebundenen Mittel deutlich unter den verfügbaren Mittel lagen. Die Aufträge wurden vom Releasemanagement begleitet und schließlich auch abgenommen. Die Ergebnisse sind in die aktuellen Releases von Kitodo.Production (3.4.3) und Kitodo.Presentation (4.0.0) eingegangen und damit allgemein verfügbar.

Von den 47.000 € wurden ca. 41.000 € in der ersten Runde verausgabt. Die verbliebenen Mittel gingen vollständig in die nächste Runde des Entwicklungsfonds ein, die im Januar 2022 nach positiver interner Evaluation mit dem neuen Aufruf zur Benennung von Entwicklungsvorschlägen unverändert starten konnte.

## 5. Fazit

In Bezug auf die Etablierung des Entwicklungsfonds für Kitodo lässt sich ein durchweg positives Fazit ziehen. Auf der konkreten Ebene der Durchführung konnten die von der Community ausgewählten Issues mit den finanziellen Mitteln des Fonds alle fristgerecht innerhalb eines Jahres und damit

rechtzeitig vor dem zweiten Durchlauf des Prozesses umgesetzt werden. An der Ideenfindung haben sich zahlreiche Teilnehmende der Community beteiligt und der Auswahlprozess fand transparent und zur Zufriedenheit aller Beteiligten statt. Nicht zuletzt konnten über den Fonds bereits im ersten Durchlauf die gewünschten finanziellen Mittel akquiriert werden.

Viel wichtiger neben dem erfolgreichen konkreten Durchlauf des Prozesses ist aber, dass auch die identifizierten Herausforderungen des Alterungsprozesses von Open Source Software erfolgreich adressiert werden konnten. Zentrale Issues der Wartung des technischen Kerns konnten umgesetzt werden – so ein notwendiges Update der Java-Version, ein Update der Elasticsearch-Engine und ein Update der TYPO3-Version. Die Mitglieder der Community sind bei der Wahl der Entwicklungsvorhaben den Vorschlägen des Releasemanagements weitgehend gefolgt und haben fast ausschließlich Entwicklungsaufgaben zur Vermeidung der Alterung der Software ausgewählt. Die Verfügbarmachung zentraler Ressourcen hat maßgeblich dazu beigetragen, dass diese für einzelne Mitglieder eher unattraktiven Aufgaben trotzdem angegangen werden konnten.

Außerdem hat die explizite Bereitstellung finanzieller Mittel für diese Form von Pflegeaufgaben mehrere Dienstleistungsunternehmen motiviert, diesen Aufgaben auch nachzukommen. Das Setzen finanzieller Anreize zur Erhöhung der extrinsischen Motivation hat zu dem gewünschten Erfolg geführt.

Alles in allem scheint die Etablierung eines zentralen Finanzierungstopfs, der in einem offenen, transparenten und partizipativen Prozess Mittel für die Bearbeitung von unattraktiven aber notwendigen Wartungsarbeiten bereitstellt, ein Weg zu sein, der auch für andere Open Source Communities interessant sein dürfte. Ob jetzt in Form eines Entwicklungsfonds, bereitgestellt durch einen Verein oder in weniger formaler Form, mag dahingestellt sein. Mit dem Entwicklungsfonds hat Kitodo ein Modell entwickelt, dass sich als best practise etablieren kann.

Trotzdem dürfte gerade für die Open-Source-Entwicklung im Bereich des öffentlichen Dienstes auch der Nachweis, dass diese Form der Umsetzung allen Regeln der Verwendung von öffentlichen Mitteln entspricht, ebenfalls eine wichtige Erkenntnis sein. Es gibt also keine formalen oder organisatorischen Hürden, die den Aufbau einer solchen Entwicklungsförderung auch in anderen Communities freier Bibliothekssoftware verhindern.

## Literaturverzeichnis

- Bonte, Achim: Chancen und Fallstricke offener, kooperativer Softwareentwicklung. Das Beispiel Kitodo, in: Bonte, Achim; Rehnolt, Juliane (Hg.): Kooperative Informationsinfrastrukturen als Chance und Herausforderung, Berlin 2018, S. 182–192. Online: <<https://doi.org/10.1515/9783110587524-022>>.
- Brügge, Bernd; Harhoff, Dietmar; Picot, Arnold u.a.: Open-Source-Software. Eine ökonomische und technische Analyse, Berlin 2004 (Springer eBook Collection Business and Economics). Online: <<https://doi.org/10.1007/978-3-642-17024-9>>.

- Coad, Peter; Lefebvre, Eric; DeLuca, Jeff: Java modeling in color with UML. Enterprise components and process; [enrich the content of your models, become a better modeler by example, deliver frequent, tangible, working results, Upper Saddle River, NJ 1999.
- Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019. Online: <<https://doi.org/10.5281/zenodo.6779479>>.
- Finck, Matthias: Usability-Engineering in der Open-Source-Softwareentwicklung, Zugl.: Hamburg, Univ., Diss., 2007, Sierke, Göttingen 2007.
- Hars, Alexander, & Ou, Shaosong (2002). Working for Free? Motivations for Participating in Open-Source Projects, in: International Journal of Electronic Commerce 6 (3), 2002, S. 25–39. Online: <<http://www.jstor.org/stable/27751021>>.
- Hippel, Eric von; Krogh, Georg von: Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science, in: Organization Science 14 (2), 2003, S. 209–223. Online: <<https://doi.org/10.1287/orsc.14.2.209.14992>>.
- Huber, Kathrin; Strötgen, Robert: Und nun? Der Weg vom Projekt zur kontinuierlichen Produktentwicklung, in: Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019, S. 72–80. Online: <<https://doi.org/10.5281/zenodo.6779479>>.
- Meyer, Sebastian; Huber, Kathrin: Konsequent Modular – ein offenes, modernes Architekturkonzept, in: Finck, Matthias; Hermann, Elena (Hg.): Abschlussbericht Kooperative Weiterentwicklung der quelloffenen Digitalisierungssoftware Kitodo.Production, Elmshorn 2019, S. 40–49. Online: <<https://doi.org/10.5281/zenodo.6779479>>.
- Vassallo, Carmine; Grano, Giovanni; Palomba, Fabio u. a.: A large-scale empirical exploration on refactoring activities in open source software projects, in: Science of Computer Programming 180, 2019, S. 1–15. Online: <<https://doi.org/10.1016/j.scico.2019.05.002>>.
- Wolf, Henning; Bleek, Wolf-Gideon: Agile Softwareentwicklung. Werte, Konzepte und Methoden, Heidelberg 2011.