ÖGD

Geschäftsstelle
der UAG DIGITALISIERUNG

WHITEPAPER

ZUM EINSATZ VON OPEN-SOURCE-SYSTEMEN IM ÖFFENTLICHEN GESUNDHEITSDIENST

Projekt:
Inhaltliche Erweiterung der Geschäftsstelle
der UAG Digitalisierung der AG ÖGD der AOLG

Projektnummer:
BE-P512LM20E51

Programm:
Förderung von Maßnahmen zur Steigerung
und Weiterentwicklung des digitalen Reifegrades
des Öffentlichen Gesundheitsdienstes in
Deutschland

Funded by
the European Union
NextGenerationEU

ELFA
EIN LAND FÜR ALLE

VORWORT

Open Source Software (OSS) ist aus der heutigen Software Landschaft nicht mehr weg zu denken. Sie hat in praktisch allen Bereichen Einzug gehalten und ist teilweise sogar Bestandteil proprietärer kommerzieller Software.

Da OSS in der Regel kostenfrei ist, stellt sie potenziell eine günstige Alternative zu proprietärer Software dar.

Gerade vor dem Hintergrund knapper Kassen, auch im Gesundheitssektor sind Überlegungen neue Software-Entwicklung auf der Basis von OSS zu realisieren daher eine valide Überlegung.

OSS heißt jedoch nicht Lizenzfrei. Jede OSS-Komponente besitzt eigene Regeln und Lizenzmodelle, welche zu berücksichtigen sind. Dabei haben sich zwar bestimmte Standard-Lizenzmodelle etabliert (z. B. GNU oder GPL), jedoch ist jeder Urheber im Prinzip frei, was die Gestaltung der Lizenzierung angeht. Dies führt mitunter zu Recht individuellen Modellen, wie der Bier-Lizenz, bei der man dem Entwickler ein Bier ausgeben muss, wenn man ihn trifft. Solche „Spaß-Lizenzen“ stellen eine Besonderheit des Thema OSS dar.

Rund um das Thema OSS haben sich zudem eine Reihe kommerzieller Geschäftsmodelle etabliert. Diese professionalisieren den Support oder bieten zusätzliche Funktionalitäten an. Allerdings führen sie zu Kosten, welche den Kostenvorteil von OSS reduzieren können.

Die Eigenentwicklung und der Eigenbetrieb von OSS erfordern zudem spezifisches Wissen, über welches nicht jeder verfügt. Daher ist eine genaue Prüfung beim Einsatz von OSS erforderlich, um das bestmögliche Ergebnis zu erzielen und nach den eigenen Bedürfnissen und Fähigkeiten auszurichten.

Das nachfolgende Positionspapier soll Nicht-IT-lern in Bereich des Gesundheitswesens als Orientierungshilfe bei der Beurteilung von OSS als möglicher Lösungsansatz bei der Entwicklung von Fachanwendungen dienen.

Eine wichtige Säule des Projekts ist die Zusammenarbeit mit relevanten Stakeholdern, um sicherzustellen, dass der Leitfaden praxisnah und auf die realen Bedürfnisse der ÖGD-Mitarbeitenden zugeschnitten ist. Dies wird durch eine Arbeitsgruppe erreicht, die sich aus Vertretern verschiedener Gesundheitsämter, strategischen IT-Planern, Sachbearbeitern sowie weiteren Interessierten zusammensetzt. Der Austausch von Erfahrungswerten und vorhandenen Bewertungen soll sicherstellen, dass alle relevanten Dimensionen abgedeckt werden und die Mitarbeitenden den tatsächlichen Wissensstand aus der Praxis einbringen.

Inhaltsverzeichnis

	Vorwort	2
1	Positionspapier OSS im ÖGD	5
1.1	Interviews und Anforderungserhebung	5
1.2	Vorteile von OSS im Gesundheitsbereich	6
1.2.1	Kosteneffizienz und langfristige Einsparungen	6
1.2.2	Sicherheit	9
1.2.3	Transparenz	10
1.2.4	Anpassungsfähigkeit an lokale Anforderungen	12
1.2.5	Förderung von Interoperabilität	12
1.3	Rahmenbedingungen, Herausforderungen und Hemmnisse	12
1.3.1	Fachkräftemangel und fehlende IT-Ressourcen	12
1.3.2	Kurze Projektzyklen und vergaberechtliche Hürden	14
1.3.3	Verantwortungs- und Haftungsfragen bei OSS	15
1.3.4	Akzeptanzprobleme bei Mitarbeitenden	16
1.3.5	Wissens- und Schulungsbedarf	17
1.3.6	Gängige Open-Source-Lizenzen und ihre rechtlichen Implikationen	18
1.4	Strategie zur Förderung von OSS im öffentlichen Sektor in Deutschland	20
1.5	Fazit	22
2	Leitfaden	23
2.1	Grundlagen	23
2.1.1	Übersetzung der Anforderungen	23
2.1.2	Arten der Anforderungen	24
2.1.3	Arten von OSS	25
2.1.4	Definition der eigenen Rolle hinsichtlich OSS	26
2.1.5	Arten von Rollen	26
2.1.6	Umsetzung und Beschaffung	27
2.2	Ein Vorgehensmodell zur praktischen Entscheidungsfindung	30
2.2.1	Was brauche ich?	30
2.2.2	Wie beschaffe ich?	34
2.2.3	Wie setze ich um?	37

2.2.4 Zusammenfassung: Ein strukturierter Leitfaden zur Umsetzung 42

2.3 Nachnutzung von OSS-Lösungen im Rahmen von SaaS 42

2.3.1 SaaS im Kontext der aktuellen Herausforderungen der Gesundheitsämter 43

2.3.2 Wertbeiträge zur Bewältigung der Herausforderungen durch SaaS 43

2.3.3 Fazit zur Nachnutzung von Open-Source Lösungen im Rahmen SaaS 44

2.4 Transformation bestehender Systeme zu Open Source 45

3 Schlusswort 46

4 Anhang 47

4.1 Umfrage 47

4.2 Quellen 49

4.3 Glossar 50

Impressum 52

1 POSITIONSPAPIER OSS IM ÖGD

Open-Source-Software (OSS) gewinnt im öffentlichen Sektor zunehmend an Bedeutung. Dennoch steckt ihr Potenzial noch in den Kinderschuhen. Während viele Behörden OSS zwar für grundlegende IT-Infrastrukturen wie Server und Web-Technologien einsetzen, ist sie in Fachverfahren noch nicht weit verbreitet. Häufig fehlen klare Strategien, Ressourcen oder das nötige Fachwissen, um OSS konsequent als Alternative zu nutzen. Gleichzeitig wächst das Bewusstsein für digitale Souveränität, und immer mehr Menschen setzen sich dafür ein, unabhängiger von proprietären Lösungen zu werden.

Digitale Herausforderungen in Gesundheitsämtern

Gesundheitsämter stehen vor enormen digitalen Herausforderungen. Sie müssen große Datenmengen sicher verarbeiten, Informationen mit anderen Behörden austauschen und gleichzeitig Datenschutzvorgaben einhalten. Viele IT-Systeme sind jedoch veraltet und lassen sich nur schwer an aktuelle Anforderungen anpassen. Die COVID-19-Pandemie hat gezeigt, dass viele dieser Systeme nicht flexibel genug sind, um schnell auf Krisensituationen reagieren zu können. Zudem fehlt es an IT-Fachkräften, die dringend benötigt werden, um moderne Technologien einzuführen und zu betreuen. Eine weitere Herausforderung: Unterschiedliche Softwarelösungen und fehlende Schnittstellen erschweren die Zusammenarbeit zwischen den Ämtern.

Einsatz aktueller Softwarelösungen

Momentan nutzen Gesundheitsämter eine Mischung aus proprietären und individuell angepassten Softwarelösungen. Einheitliche Standards fehlen oft, weshalb viele Ämter eigene, maßgeschneiderte Systeme verwenden. Das mag zunächst praktisch erscheinen, führt aber langfristig zu Problemen – etwa hohen Wartungskosten und fehlender Kompatibilität mit neuen Technologien. Cloudbasierte Lösungen werden zwar immer häufiger eingesetzt, doch bestehen hier oft Bedenken hinsichtlich des Datenschutzes. OSS wird bereits in einigen Bereichen genutzt, zum Beispiel für Datenanalysen und digitale Kommunikation, sie ist aber noch nicht die Regel.

Verfügbarkeit und Schwächen proprietärer Systeme

Viele Gesundheitsämter setzen nach wie vor auf proprietäre Software, obwohl diese einige Herausforderungen mit sich bringt. Hohe Lizenzkosten, eingeschränkte Anpassungsmöglichkeiten und die Abhängigkeit von wenigen Anbietern erschweren es, flexibel zu bleiben. Oft sind diese Systeme nicht speziell für die Bedürfnisse der Gesundheitsämter entwickelt und müssen aufwendig angepasst werden. Noch schwieriger wird es, wenn proprietäre Lösungen keine offenen Schnittstellen bieten – dann wird der Austausch von Daten mit anderen Behörden zum Kraftakt. OSS könnte hier eine langfristige, nachhaltige Alternative sein: Sie ist flexibel, anpassbar und bietet die Möglichkeit, digitale Souveränität zu stärken.

1.1 Interviews und Anforderungserhebung

Die Befragungen und Interviews verdeutlichen, dass die Digitalisierung im öffentlichen Gesundheitswesen auf zahlreiche Herausforderungen trifft, aber auch Chancen bietet. Eine zentrale Erkenntnis ist, dass die Beteiligung an Open-Source-Projekten vor allem durch den Wunsch nach Netzwerkbildung, der Suche nach passender Lösungen für die eigene Behörde sowie dem

Austausch über Best Practices motiviert wird. Viele Beteiligte sehen OSS nicht nur als technische Lösung, sondern auch als Möglichkeit, voneinander zu lernen und durch Kooperationen effizientere Strukturen zu schaffen.

Erfahrungen mit den Anforderungen und Prozessen im Gesundheitssektor sind ein entscheidender Faktor für eine erfolgreiche Digitalisierung. Die Befragten verfügen über wertvolles Wissen darüber, welche spezifischen Herausforderungen in ihrem Bereich bestehen und welche Lösungen tatsächlich praxistauglich sind. Daraus ergibt sich die Notwendigkeit, bestehende Anforderungen systematisch zu erfassen und zielgerichtet in digitale Lösungen zu übersetzen.

Ein wesentliches Hindernis für die aktive Teilnahme an digitalen Projekten sind zeitliche Einschränkungen. Viele Stakeholder können sich nicht ausschließlich synchron in Meetings einbringen, sondern benötigen auch Möglichkeiten zur asynchronen Mitarbeit. Um ihnen die Teilnahme zu erleichtern, sollten Arbeitsprozesse so gestaltet werden, dass sie flexibel und unabhängig von festen Terminen genutzt werden können. Zudem variieren die technischen Vorkenntnisse stark – während einige Akteure bereits über Erfahrung mit digitalen Systemen verfügen, benötigen andere eine niedrigschwellige Einführung. Die Inhalte und Formate digitaler Veranstaltungen müssen daher dem unterschiedlichen Kenntnisstand angepasst werden.

Besonders wichtig sind den Beteiligten eine enge Kooperation und ein intensiver Wissensaustausch. Neben der Entwicklung neuer Lösungen liegt das Interesse auch auf der Beschaffung und Implementierung geeigneter Technologien. Hierbei wünschen sich viele eine zentrale Anlaufstelle für fachliche Fragen sowie leicht zugängliche Informationen zum Thema OSS. Die Bereitstellung von verständlichen und praxisnahen Ressourcen würde die Akzeptanz und Nutzbarkeit von OSS erheblich erleichtern.

Abschließend zeigt sich, dass eine erfolgreiche Digitalisierung nicht nur technologische Fortschritte, sondern auch gut durchdachte Konzepte für Wissensvermittlung, Zusammenarbeit und nachhaltige Implementierung erfordert. Die Zentralisierung von Wissen und Kompetenzstellen könnte dabei helfen, Fachkräfte zu entlasten und den Einsatz digitaler Lösungen effizienter zu gestalten.

Weitere Details zu den Befragungsergebnissen sind im Anhang zu finden.

1.2 Vorteile von OSS im Gesundheitsbereich

1.2.1 Kosteneffizienz und langfristige Einsparungen

Open-Source-Software (OSS) bietet das Potenzial, Lizenzkosten zu vermeiden und Entwicklungskosten durch gemeinschaftliche Arbeit zu teilen. Dies kann insbesondere dann zu Einsparungen führen, wenn mehrere Institutionen gemeinsam an Lösungen arbeiten oder bestehende OSS-Komponenten nachgenutzt werden.

Gleichzeitig ist zu beachten, dass Einsparpotenziale nicht automatisch entstehen. Implementierung, Anpassung an lokale Gegebenheiten, Integration in bestehende Systemlandschaften sowie Wartung und Weiterentwicklung erfordern in der Regel qualifiziertes Personal. Gerade in kleineren Gesundheitsämtern können diese Aufgaben nicht immer aus eigener Kraft erbracht werden. Es ist daher essenziell, den zusätzlichen Aufwand realistisch zu kalkulieren – insbesondere auch im Hinblick auf Schulung, Betreuung, Betrieb und Support.

Opportunitätskosten – wie der zeitliche Aufwand von Fachpersonal oder die Einschränkungen durch begrenzte technische Ressourcen – müssen ebenso in die Wirtschaftlichkeitsbetrachtung einbezogen werden. Die Annahme, dass OSS per se kostengünstiger sei, greift daher zu kurz. Wirtschaftlichkeit hängt stark von den Rahmenbedingungen vor Ort, vorhandenen Kompetenzen sowie der strategischen Zusammenarbeit mit Dienstleistern oder anderen Ämtern ab.

Folgende Aspekte sind im Zusammenhang mit OSS zu beachten:

Direkte Kostenvorteile

Open-Source-Software ermöglicht es Organisationen, ohne Lizenzkosten auf leistungsfähige Softwarelösungen zuzugreifen. Dies reduziert die langfristigen Betriebskosten erheblich und vermeidet eine Abhängigkeit von einzelnen Anbietern (Vendor-Lock-in). Zusätzlich profitieren viele Institutionen von gemeinschaftlicher Entwicklung und geteilten Weiterentwicklungskosten, wodurch finanzielle Aufwendungen breiter verteilt werden können. Ein weiterer Vorteil liegt in der hohen Anpassungsfähigkeit von OSS, die es ermöglicht, maßgeschneiderte Lösungen für spezifische Bedürfnisse zu entwickeln, ohne auf kostenpflichtige Erweiterungen proprietärer Anbieter angewiesen zu sein.

Folgende mögliche Kosten sind im Zusammenhang mit OSS zu beachten:

Kosten für professionellen Support durch einen Dienstleister (externe Kosten)	Rund um OSS haben sich eine Reihe von Geschäftsmodellen etabliert, von denen eines davon der professionelle Support für OSS darstellt. Beispiele hierfür sind Wartungsverträge für Linux, MySQL oder Java Derivate durch Hersteller wie Oracle, Suse oder Red Hat. Diese können sowohl nutzungsabhängig (beispielsweise Zahl der Instanzen), nach Größe der Organisation oder pauschal geregelt sein. Die Kosten für einen solchen Support, sofern erforderlich oder gewünscht, sollten daher im Vorhinein geklärt werden und in die Gesamtkostenbetrachtung einfließen.
Kosten für Premium Features / Kosten-pflichtige Zusatz-funktionen (externe Kosten)	Ein weiteres Geschäftsmodell rund um OSS geht mit der kostenpflichtigen funktionalen Erweiterung von OSS einher. Das bedeutet, die eigentliche Kernsoftware ist weiterhin OSS und kostenfrei, jedoch können hilfreiche Ergänzungen der Software kostenpflichtig sein und damit unter getrennter proprietärer Lizenz stehen. Die Nutzung solcher zusätzlichen Funktionen kann sich im Laufe des Entwicklungsprozesses ergeben, in dessen Rahmen der funktionale Bedarf im Vordergrund steht.
Personalaufwand für Community-Engagement (interner Aufwand)	Je nachdem, wie intensiv das Engagement innerhalb einer OSS-Community sein soll, kann dies unterschiedlich viele Personalressourcen erfordern. Die aktive Teilnahme an einer Community ist ein zentraler Bestandteil von OSS und unterscheidet sich deutlich von proprietären Lösungen, bei denen eine solche Form der Zusammenarbeit meist nicht existiert. Damit die Verwaltung OSS effizient nutzen und weiterentwickeln kann, ist es essenziell, sogenannte Power-User zu identifizieren. Diese fungieren als Bindeglied zwischen Fachbereich und IT, indem sie Anforderungen beschreiben, Verbesserungspotenziale erkennen und den Weiterentwicklungsbedarf der Software gezielt adressieren. Power-User sollten nicht nur in der Nutzung der Software geschult sein, sondern auch ein grund-legendes Verständnis für technische Möglichkeiten und Community-Prozesse besitzen. Die Mitwirkung in einer OSS-Community fällt zusätzlich in den Rahmen der eigenen Rollen-definition. Da der genaue Ressourcenaufwand zu Beginn schwer abzuschätzen ist und sich mit Aufgaben der Software-Entwicklung sowie denen eines Open Source Program Office (OSPO) überschneiden kann, sollte zunächst eine Überschlagspauschale für den Personalaufwand angesetzt werden. Diese kann im Laufe der Zeit anhand praktischer Erfahrungswerte präzisiert werden. Eine frühzeitige Identifikation und Befähigung von Power-Usern kann dazu beitragen, dass OSS-Projekte in der öffentlichen Verwaltung effektiver gesteuert werden und eine nachhaltige Weiterentwicklung gewährleistet ist.

Software-Pflege (interner / externer Aufwand)	Aufgrund der hohen Gestaltungsflexibilität von OSS, insbesondere in der Interaktion mit weiterer Software, kann sich beispielsweise die dauerhafte Pflege, Anpassung und Sicherstellung von Kompatibilitäten im Vergleich zu proprietärer Software aufwändiger gestalten. Hierzu sollte eine initiale Abschätzung erfolgen, welche aufgrund von Erfahrungswerten im Laufe der Zeit präzisiert werden kann.
OSPO (interner Aufwand)	Aufgrund der spezifischen Anforderungen von OSS in den Bereichen Entwicklung, Lizenzierung, Community-Management und Governance kann die Einrichtung eines OSPO sinnvoll sein. Ein OSPO übernimmt die Koordination aller OSS-Aktivitäten – sowohl intern als auch extern – und kann strategische sowie operative Aufgaben wahrnehmen. Dazu gehört die Definition einer OSS-Governance sowie die Entwicklung einer übergeordneten OSS-Strategie. Der Themenbereich Lizenzierung kann dabei als eigenständiger Schwerpunkt entweder innerhalb des OSPO oder in enger Zusammenarbeit mit dem klassischen Lizenzmanagement verankert werden. Allerdings muss die Positionierung eines OSPO sorgfältig überdacht werden, da der damit verbundene Aufwand und die notwendige Expertise nicht in jeder Organisation wirtschaftlich tragfähig sind. In der Regel lohnt sich die Einrichtung eines OSPO insbesondere bei einem zentralen IT-Dienstleister oder einer Organisation mit breiter Softwareentwicklungskompetenz sowie bei einer Behörde mit übergreifender Digitalisierungsverantwortung, wie einem Digitalisierungsministerium oder einer zentralen IT-Strategieeinheit. In kleineren Organisationen oder Fachbehörden kann es effizienter sein, OSS-Aktivitäten durch bestehende Strukturen, etwa das IT-Management oder das Lizenzmanagement, steuern zu lassen. Alternativ kann auch eine Kooperation mit bereits etablierten OSPOs in anderen Behörden oder übergreifenden Institutionen eine sinnvolle Lösung sein. Der Umfang eines OSPOs sollte sich daher nach dem spezifischen Bedarf, den Governance-Anforderungen und der Organisationsstruktur richten und anteilig in die Kostenbetrachtung von OSS einfließen.

Tabelle 1: OSS Kostenarten (fortgesetzt)

Warum sich OSS trotzdem lohnt

Trotz dieser möglichen Zusatzkosten bleibt OSS in vielen Fällen eine wirtschaftliche Alternative, insbesondere wenn langfristige Einsparungen durch Lizenzfreiheit, gemeinschaftliche Weiterentwicklung und reduzierte Herstellerabhängigkeit in Betracht gezogen werden. Auch proprietäre Software erfordert Wartung, Support und Zusatzmodule – oft zu höheren Kosten und mit geringerer Flexibilität als OSS.

Merke:

Die Nutzung von Open-Source-Software lässt sich mit dem Anlegen eines eigenen Gemüsegartens vergleichen. Während man bei proprietärer Software fertige Lösungen „aus dem Supermarkt“ kauft und von Preisgestaltung und Verfügbarkeit des Anbieters abhängig ist, investiert man bei Open Source zunächst eigene Ressourcen in Pflege und Weiterentwicklung. Auch wenn die Nutzung anfangs kostengünstig erscheinen mag, muss Zeit für Pflege, Gestaltung und Instandhaltung eingeplant werden. Für OSS gilt ähnliches – ohne kontinuierliches Engagement entstehen keine nachhaltigen Lösungen.

1.2.2 Sicherheit

Open-Source-Software (OSS) wird mitunter als besonders sicher eingeschätzt – unter anderem deshalb, weil der Quellcode offen einsehbar ist und somit viele Augen potenzielle Sicherheitslücken frühzeitig erkennen können. In der Praxis hängt die tatsächliche Sicherheit jedoch stark davon ab, wie sorgfältig eine Software gepflegt, überwacht und aktualisiert wird.

Der Einsatz von OSS überträgt mehr Verantwortung auf die betreibende Organisation: Sicherheitsupdates müssen aktiv beobachtet, Sicherheitslücken bewertet und Maßnahmen zur Absicherung zeitnah umgesetzt werden. Dies erfordert entsprechende Prozesse, Monitoring-Mechanismen und qualifiziertes Personal.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) weist in diesem Zusammenhang darauf hin: „Die Verantwortung für die Sicherheit ist bei OSS jedoch umfangreicher und muss kontinuierlich überwacht und verbessert werden."

Ob eine OSS-Lösung am Ende sicherer oder weniger sicher ist als eine proprietäre Lösung, hängt weniger von der Lizenzform ab – sondern vielmehr davon, ob es gelingt, angemessene organisatorische und technische Sicherheitsstandards zu etablieren und aufrechtzuerhalten.

Sicherheit ist hierbei unter mehreren Aspekten zu betrachten:

Betriebssicherheit	Stabilität und Performance der Software im Rahmen der etablierten Betriebsverfahren.
IT-Sicherheit	Sicherheit gegenüber Cyberangriffen (siehe CRA – Cyber-Resilienz Act der EU).
Datenschutz	Sicherheit und Schutz personenbezogener Daten entsprechend DSGVO.

Tabelle 2: OSS Sicherheitsaspekte

Merke:

Open Source Software ist nicht unsicherer als proprietäre Software – im Gegenteil: Die Transparenz des Quellcodes ermöglicht eine unabhängige Prüfung und schnelle Reaktion auf Sicherheitslücken, wodurch OSS in vielen Fällen sogar sicherer sein kann. Dies sieht auch das Bundesamt für Sicherheit in der Informationstechnik (BSI) so. Laut BSI kann OSS durch offene Codeprüfung und Community-Beteiligung hohe Sicherheitsstandards gewährleisten, sofern bewährte Entwicklungs- und Betriebsprinzipien beachtet werden. Entscheidend für die Sicherheit sind die Beachtung bewährter Entwicklungs- und Betriebspraktiken sowie das Know-how der beteiligten Personen. Die Verantwortung für die Sicherheit ist bei OSS jedoch umfangreicher und muss kontinuierlich überwacht und verbessert werden.

Quellen: Bundesamt für Sicherheit in der Informationstechnik (BSI), „Freie Software – Open Source Software und IT-Sicherheit", abrufbar unter: bsi.bund.de

Alle drei Aspekte sind insbesondere für Anwendungen im Gesundheitssektor von besonderer Bedeutung. Die Einhaltung bestimmter Entwicklungs- und Betriebskriterien als wesentliches Element der Sicherheit von OSS spielt somit eine zentrale Rolle. Durch das Einspielen regelmäßiger Software-Updates, den Bezug der Software aus vertrauenswürdigen Quellen sowie die Einhaltung bestimmter Entwicklungsprinzipien können diesbezügliche Risiken auf einem vergleichbaren Niveau wie bei proprietärer Software gehalten werden.

Darüber hinaus könnten Bedenken hinsichtlich der Veröffentlichungspflichten aufgrund eines Copyleft-Effekt bestehen, welche die Software angreifbar machen. Da jedoch in der Regel nur Teile einer Software dem Copyleft-Effekt unterliegen, kann dieses Risiko eingeschränkt werden, indem Software (oder Komponenten wie Libraries) für bestimmte Anwendungsteile bewusst ausgewählt werden und somit kontrollierbar ist, was veröffentlicht werden muss und was nicht.

Die Einrichtung eines Open Source Program Office (OSPO) kann zusätzlich zur Verbesserung der Rechtssicherheit beitragen, eine aktive Beteiligung an der OSS-Community fördern und die Kommunikation sowie wirtschaftliche Aspekte verbessern. Falls ein OSPO nicht eingerichtet werden kann, können bestimmte Aufgaben zumindest in das Lizenzmanagement integriert werden. Dies erfordert jedoch spezifisches Fachwissen, um rechtliche und sicherheitsrelevante Anforderungen zuverlässig zu erfüllen.

1.2.3 Transparenz

Transparenz ist im Bereich OSS mindestens genauso wichtig wie beim Einsatz proprietärer Software. Dies betrifft eine ganze Reihe unterschiedlicher Formen der Transparenz, welche Voraussetzung für eine aktive Steuerung des Einsatzes von OSS sind. Sie alle haben spezifischen Besonderheiten und Ausprägungen bezüglich OSS:

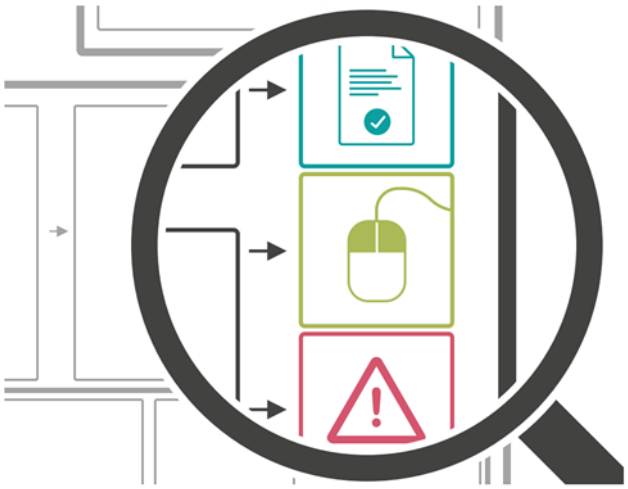
Kostentransparenz	Kostentransparenz ist sowohl für die jährliche Budget- und Haushaltsplanung als auch für die Argumentation gegenüber den Verfechtern proprietärer Lösungen essenziell. Bei SaaS-Lösungen, die in der Regel über eine nachgelagerte nutzungsabhängige Abrechnung verfügen und daher nur schwer planbar sind, gewinnt das Thema Kostentransparenz zusätzlich an Bedeutung. Bei OSS ist das Herstellen von Kostentransparenz oft schwieriger als bei proprietärer Software, weil zwar keine Anschaffungskosten anfallen, aber versteckte externe und interne Kosten entstehen können (etwa für Wartung oder Bereitstellung von Ressourcen für Community-Engagement).
Nutzungstransparenz	Die Nutzungstransparenz ist unter anderem Voraussetzung für die spätere Sicherstellung der Lizenzcompliance. Sie kann zudem Voraussetzung für die Kostentransparenz und Planbarkeit sein. Insbesondere bei SaaS-Services ist eine Beeinflussung der Kosten in der Regel nur über die Nutzung möglich, weshalb die Nutzungstransparenz essenziell für die Kontrolle und Planbarkeit der Kosten ist. Im Vergleich zu proprietärer Software ist die Nutzungstransparenz bei OSS schwieriger herzustellen, da OSS häufig unkontrolliert und dezentral eingebunden wird – insbesondere durch Entwickler, die frei verfügbare Bibliotheken nutzen, ohne einen zentralen Freigabeprozess zu durchlaufen. Zudem weisen OSS-Komponenten oft zahlreiche transitive Abhängigkeiten auf, die automatisch mitinstalliert werden und nur schwer zu überblicken sind.
Lizenztransparenz (Lizenzcompliance)	Im Gegensatz zu proprietärer Lizenzierung, welche zumeist auf der Lizenzierung nach Nutzung aufsetzen, ist die Lizenzierung im Bereich OSS in der Regel qualitativer Art. Das heißt es müssen bestimmte Formalien wie die Veröffentlichungspflichten im Rahmen von Copyleft eingehalten werden. Eine Messung der Nutzung zum Zweck der Lizenzierung findet in der Regel nicht statt. Dies trifft auch für SaaS Services von OSS zu. Das bedeutet, die Kosten eines SaaS Services resultieren zumeist aus Betriebs und Supportkosten und nicht aus Lizenzkosten. Jenseits der Lizenzierung kann jedoch eine Nutzungsmessung aus Kostengründen erfolgen. Zu beachten sind dennoch Wechselwirkungen zwischen den Lizenzmodellen unterschiedlicher OSS-Komponenten innerhalb einer Anwendung. Diese können sich gegebenenfalls widersprechen und somit problematisch sein. Das heißt eine Prüfung der Lizenzierung einzelner Software oder Komponenten ist im Rahmen des Entwicklungsprozesses unbedingt erforderlich. Auch wenn im Bereich OSS die Wahrscheinlichkeit von Lizenzaudits geringer ist als bei proprietärer Software, können diese nicht ausgeschlossen werden. Beispiele liegen hier im Bereich Java und der Firma Oracle. Daher sollte auch die Lizenzcompliance von OSS nicht vernachlässigt werden.

Risikotransparenz	Hierbei ist unter anderem das Thema Schwachstellenmanagement zu betrachten, welchem aufgrund der individuellen Entwicklungsmöglichkeiten bei OSS und im Rahmen der IT-Sicherheitsanforderungen eine besondere Bedeutung zukommt. Gerade, weil OSS häufig in komplexen Software-Stacks eingesetzt wird, wodurch unentdeckte Schwachstellen erhebliche Auswirkungen haben können, ist ein effektives Schwachstellenmanagement notwendig, um Sicherheitslücken frühzeitig erkennen und beheben zu können.
Transparenz über Softwarekomponenten via SBOM	SBOM ist eine Art „Stückliste“ für eine Anwendung und listet deren Komponenten ungeachtet OSS oder proprietär) auf. Dabei fundiert sie auf einem Datenstandard CycloneDX, SPDX oder auch BSI – Technische Richtlinie Cyber-Resilienz-Anforderungen / TR-03183), welcher verschiedene Datenfelder (beispielsweise „Lizenz”) beinhaltet. Weitere Felder sind unter anderem Hersteller, Komponentenname, Komponentenversion oder Abhängigkeit von anderen Komponenten). Bei der Erstellung eigener Lösungen müssen zudem eigene SBOM erstellt und gepflegt werden. SBOM werden unter anderem in Bezug auf Lieferkettenverfolgung (Lieferkettengesetz) oder IT-Sicherheit (Cyber-Resilienz-Verordnung / (EU) 2024 / 2847) angewandt. Daher ist die Verwaltung von SBOM oft in der IT-sicherheit angesiedelt. Es gibt jedoch keinen organisatorisch durch Standards fix definierten Ort an dem diese standfinden muss, sodass die Verwaltung der SBOM je nach Bedarf angesiedelt werden kann. SBOM ist gerade bei OSS entscheidend, da OSS häufig aus vielen externen Abhängigkeiten besteht, deren Sicherheitsrisiken und Lizenzanforderungen überwacht werden müssen. Ohne SBOM ist es kaum möglich, Schwachstellen oder Lizenzverstöße zu erkennen, was rechtliche und sicherheitstechnische Risiken erhöht. Während bei proprietärer Software die Hersteller ihre Komponenten intern kontrollieren, fehlt bei OSS oft eine zentrale Instanz, die für Sicherheitsupdates und Compliance sorgt.

Tabelle 3: OSS Transparenz (Fortsetzung nächste Seite)

Merke:

Es gibt eine Reihe von Ausprägungen der Transparenz, welche insbesondere auf Kosten- und Nutzungskontrolle sowie Rechtssicherheit einzahlen. Die spezifischen Formen zur Erreichung der Transparenz können dabei je nach Organisation und Anforderungen variieren beziehungsweise definiert und umgesetzt werden. Eine Tool-Unterstützung ist in (fast) jedem Fall erforderlich, da OSS in der Regel nicht nur bei einer Lösung zum Einsatz kommt, sondern potenziell in vielen Bereichen einer Organisation. Daher ist eine manuelle Verwaltung von SBOM und Lizenzen in der Regel nicht möglich.



Die praktische Regelung zur Erreichung der jeweiligen Transparenz muss in Abstimmung mit den individuellen Beteiligten stattfinden. Sie kann (und muss gegebenenfalls aufgrund der Menge) zudem Tool-unterstützt sein, wobei hier gegebenenfalls eine Verzahnung mit bereits vorhandenen Tools stattfinden muss.

1.2.4 Anpassungsfähigkeit an lokale Anforderungen

Die Möglichkeit, OSS flexibel an lokale Anforderungen anzupassen, ist ein großer Vorteil – insbesondere im Vergleich zu starren, proprietären Systemen. Diese Freiheit bringt jedoch auch Verantwortung mit sich: Wer Open-Source-Komponenten verändert oder erweitert, ist im Gegenzug meist selbst für die Pflege und die technische Aktualisierung zuständig.

Insbesondere umfangreiche lokale Modifikationen können langfristig zu einem „Fork“ führen – also einer Abspaltung vom Ursprungssystem. Dies erschwert nicht nur die Integration neuer Versionen oder Sicherheitsupdates, sondern isoliert auch den Weiterentwicklungsprozess. In der Folge entstehen neue Abhängigkeiten, dieses Mal nicht gegenüber einem Hersteller, sondern gegenüber eigenem (oder beauftragtem) technischem Personal.

Dieser zusätzliche Pflegeaufwand ist häufig nicht einkalkuliert – gerade in kleineren Behörden mit begrenzten IT-Ressourcen. Deshalb sollte bei jeder Anpassung sorgfältig abgewogen werden:

- Lässt sich die Funktionalität auch über Konfiguration oder Schnittstellen realisieren?
- Gibt es andere Ämter mit ähnlichen Anforderungen, mit denen man sich abstimmen kann?
- Ist die Änderung langfristig wartbar oder führt sie zu einer isolierten Sonderlösung?

Nur wenn Anpassungen strategisch geplant und dokumentiert werden – idealerweise in enger Abstimmung mit der Community –, lässt sich die Balance zwischen lokaler Passung und überregionaler Wartbarkeit erhalten.

Merke:

Wer stark individualisiert, muss langfristig selbst pflegen – oder isoliert sich technisch. OSS lebt vom Mitmachen, aber auch vom Mitgehen mit der Weiterentwicklung.

1.2.5 Förderung von Interoperabilität

Die Verwendung von OSS fördert zudem die Interoperabilität in der IT. In der Grundstruktur von OSS ist oft die Kompatibilität mit anderen OSS-Komponenten verankert, teilweise noch in höherem Maße als bei proprietärer Software.

Auch die freie Verwendbarkeit von Softwaremodulen unter OSS für eigene Lösungen ist in OSS üblich (siehe Plattform OpenCode / GitHub). Dies wird unter anderem durch die Verwendung offener Standards erreicht, welche erheblich zur Interoperabilität von OSS beitragen.

Die Entwicklung eigener Lösungen auf Basis Interoperabilitätsziele sollte daher immer Gegenstand eigener Entwicklungen sein.

1.3 Rahmenbedingungen, Herausforderungen und Hemmnisse

1.3.1 Fachkräftemangel und fehlende IT-Ressourcen

Der Fachkräftemangel und fehlende IT-Ressourcen können eine Herausforderung bei der Konzeption, der Umsetzung und dem Betrieb von OSS-Anwendungen sein, welche bis zum Scheitern entsprechender Projekte führen kann.

Prinzipiell sollen IT-Systeme, wie die entsprechenden Fachanwendungen, zur Entlastung durch Digitalisierung und Automation fachlicher Prozesse führen. Allerdings sind auch für die Umsetzung und den Betrieb (fachliche, anwendungstechnisch und systemtechnisch) qualifizierte Personalressourcen und finanzielle Mittel (beispielsweise für Support oder Dienstleister) erforderlich.

OSS bildet hier keine Ausnahme. Im Gegenteil: Qualifizierte und erfahrene Entwickler auf dem Gebiet OSS können schwer zu finden sein, da hierzu spezifisches Know-how erforderlich ist. Darüber hinaus unterscheidet sich die Softwareentwicklung unter OSS in vielen Details von proprietären Softwareprojekten. Dies beginnt bei der Lizenzierung (qualitative anstatt quantitative Lizenzierung), geht über die Teilnahme an Communities (beispielsweise Linux, GPL oder andere) bis hin zu anderen Entwicklungs- (zum Beispiel Low Code-Frameworks) und Dokumentationswerkzeugen (beispielsweise Gitlab oder GitHub).

Sofern nicht bereits entsprechende Ressourcen intern vorhanden sind, kann die Beauftragung von Dienstleistern ein Weg sein, Ressourcen zu beschaffen. Die Bereitstellung von Ressourcen über Personalbeschaffungsmaßnahmen dauert in der Regel länger und die Bereitstellung neuer Planstellen ist oft nicht möglich, weshalb dieser Ansatz nicht immer erfolgreich ist.

Die Verfügbarkeit beziehungsweise Beschaffung entsprechender Ressourcen ist zudem unter der zeitlichen Maßgabe der Umsetzungsrahmenbedingungen zu betrachten.

Merke:

Aufgrund der spezifischen Anforderungen von OSS sind Fachkräfte mit entsprechendem Wissen erforderlich. Eine verzögerte oder fehlende Bereitstellung kann zu Projektverzögerungen oder sogar zum Scheitern führen. Schlecht qualifiziertes Personal kann gerade in Bezug auf die Einhaltung erforderlicher Entwicklungsvorgeben, wie etwa der IT-Sicherheit, zu Problemen führen.

Daher kommt der Verfügbarkeit qualifizierten Personals eine Schlüsselrolle zu. Sind keine internen Ressourcen verfügbar, können auch entsprechende Dienstleister beauftragt werden. Dies beeinfluss jedoch sowohl Entscheidungen zur Art der Entwicklung als auch gegebenenfalls der Art des Betriebs (Betriebsmodell) und ist unter vergaberechtlichen Aspekten zu betrachten.

1.3.2 Kurze Projektzyklen und vergaberechtliche Hürden

Im öffentlichen Sektor schreibt das Vergaberecht die regelmäßige Neuvergabe von Leistungen vor, um Wettbewerb und Transparenz zu gewährleisten. Diese Regelung gilt unabhängig davon, ob es sich um proprietäre Software oder OSS handelt. Auch für OSS-Lösungen sind Entwicklung, Betrieb und Support den vergaberechtlichen Anforderungen unterworfen. Mit einer geeigneten Vergabestrategie kann jedoch sichergestellt werden, dass der passende Dienstleister den Zuschlag erhält.

Ein häufig geäußerter Vorbehalt gegenüber OSS ist die potenzielle Herausforderung des Wissensverlusts bei einem Anbieterwechsel. Zwar ist der Quellcode von OSS frei zugänglich, doch projektspezifische Anpassungen und individuelle Konfigurationen erfordern oft eine umfangreiche Einarbeitung. Dieses Problem ist allerdings nicht spezifisch für OSS – auch bei proprietärer Software müssen neue Dienstleister sich erst mit der bestehenden Lösung vertraut machen. Die richtige Leistungsbeschreibung und Vertragsgestaltung können dazu beitragen, den Übergang effizient zu gestalten.

Ein Vorteil von OSS ist sogar, dass deren Beschaffung oft flexibler gestaltet werden kann. Support- und Wartungsdienstleistungen lassen sich unabhängig von der Software selbst ausschreiben, und wenn interne Ressourcen vorhanden sind, kann der Support sogar vollständig intern übernommen werden. Dadurch kann in vielen Fällen eine Ausschreibung für den Support entfallen.

Strategien für eine langfristige Vergabepraxis

Damit OSS in der öffentlichen Verwaltung erfolgreich genutzt werden kann, sollten einige Aspekte in der Vergabepraxis besonders berücksichtigt werden:

- **Langfristige Planung:** Statt kurzfristiger Ausschreibungszyklen kann eine strategische Vergabe über längere Zeiträume – beispielsweise vier Jahre oder mehr – dazu beitragen, Wissensaufbau zu sichern und Stabilität in der Entwicklung zu gewährleisten.
- **Detaillierte Leistungsbeschreibung:** Eine präzise Definition der benötigten Leistungen und Kompetenzen im Ausschreibungsverfahren stellt sicher, dass die Verwaltung ihren gewünschten Dienstleister erhält.
- **Flexibilität in der Beschaffung:** OSS ermöglicht eine schrittweise Einführung und Weiterentwicklung durch modulare Ausschreibungen einzelner Leistungen wie Entwicklung, Wartung oder Support.

Merke:

Durch eine differenzierte Vergabestrategie und das richtige Maß an Fingerspitzengefühl lassen sich wirtschaftliche, technische und strategische Vorteile von OSS in der öffentlichen Verwaltung optimal nutzen.

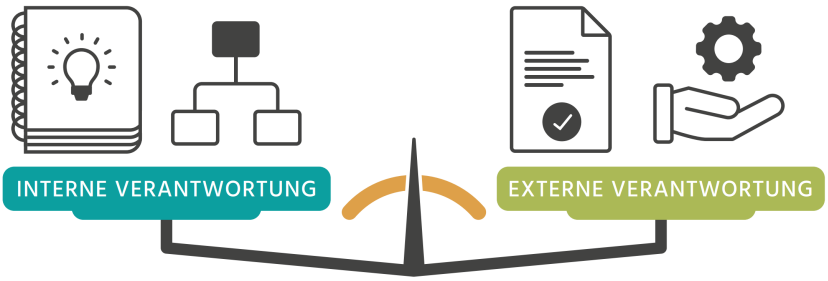
1.3.3 Verantwortungs- und Haftungsfragen bei OSS

Die Einführung von OSS in der öffentlichen Verwaltung wirft häufig Fragen zur Verantwortung und Haftung auf. Insbesondere Mitarbeitende in Behörden befürchten, persönlich für Fehlfunktionen oder Sicherheitslücken verantwortlich gemacht zu werden, wenn sie OSS einsetzen. In der Praxis lassen sich diese Bedenken jedoch durch eine klare Trennung der Verantwortlichkeiten sowie durch saubere Dokumentation und vertragliche Regelungen weitgehend ausräumen.

Klare Trennung der Verantwortung: intern und extern

Die Verantwortung für den Einsatz von OSS in der Verwaltung kann in zwei Bereiche unterteilt werden:

- Interne Verantwortung: Wenn klar festgelegt ist, welche Funktionen benötigt werden und welche Sicherheits- und Wartungskriterien erfüllt sein müssen – und diese Vorgaben auch eingehalten werden – bestehen keine rechtlichen Risiken für Mitarbeitende. Insofern ergeben sich keine Besonderheiten gegenüber dem Einsatz proprietärer Software. Der Unterschied besteht lediglich darin, dass die OSS-Landschaft komplexer und den Mitarbeitenden in der Regel weniger vertraut ist. Eine präzise Bedarfsanalyse kann jedoch Unsicherheiten minimieren und die spätere Beschaffung oder Entwicklung erleichtern. Die öffentliche Verwaltung muss dafür ihre Anforderungen an eine Softwarelösung eindeutig definieren und umfassend dokumentieren. Eine klare Rollenverteilung innerhalb der Verwaltung kann außerdem dazu beitragen, dass Aufgaben und Zuständigkeiten nachvollziehbar geregelt sind. Eine fundierte Risikoanalyse und ein gutes Risiko-management hilft, Risiken zu erkennen und geeignete Schutzmaßnahmen frühzeitig zu implementieren.
- Externe Verantwortung: Für Wartung, Support und Weiterentwicklung von OSS kann – wie bei proprietärer Software – ein Dienstleister beauftragt werden. Durch vertragliche Vereinbarungen kann sichergestellt werden, dass Haftungsfragen geklärt und Risiken minimiert werden. Insoweit ergeben sich keine Unterschiede zu proprietärer Software.



Vertragliche Regelungen mit Dienstleistern

Ein häufiges Missverständnis ist, dass OSS ohne vertragliche Absicherung genutzt werden muss, da der Quellcode offen ist. Tatsächlich können und sollten Verwaltungen, genauso wie bei proprietärer Software, Dienstleister beauftragen, um Wartung, Support und Weiterentwicklung sicherzustellen.

Durch entsprechende Haftungs- und Sachmangelhaftungsregelungen in den Verträgen mit den Dienstleistern kann für eine klare Verantwortungsverteilung gesorgt werden. Durch die Vereinbarung von sinnvollen Reaktions- und Lösungszeiten in Service Level Agreements lassen sich weiter Risiken minimieren. Durch diese vertraglichen Absicherungen bleibt die öffentliche Verwaltung handlungsfähig, ohne unkalkulierbare (Haftungs-)Risiken einzugehen.

Lizenzrechtliche Prüfungen

Unsicherheiten bezüglich Lizenzbestimmungen sind ein weiteres Hemmnis für den Einsatz von OSS. Da die verschiedene OSS-Lizenzen ganz unterschiedliche Anforderungen stellen, welche nicht nur bei der Nutzung selbst, sondern auch bei Weiterentwicklungen oder der Kombinationen mit anderer Software beachtet werden müssen (z. B. Copyleft-Klauseln oder Offenlegungspflichten), kann es leicht zu Lizenzverstößen kommen. Auch Verstöße gegen OSS-Lizenzbedingungen stellen Urheberrechtsverstöße dar, was rechtliche Konsequenzen wie Unterlassungs- und Schadensersatzansprüche sowie Reputationsschäden nach sich ziehen kann.

Dieses Risiko lässt sich jedoch dadurch minimieren, dass jede OSS-Nutzung vorab intern oder durch externe Rechtsberater geprüft wird. Durch eine gründliche rechtliche Prüfung kann sichergestellt werden, dass OSS rechtskonform genutzt wird und keine unvorhergesehenen Verpflichtungen entstehen.

Hierbei sollte vor der Einbindung einer OSS-Komponente geprüft werden, unter welcher Lizenz diese steht. Insbesondere sollte geklärt werden, ob die Lizenz zur geplanten Nutzung passt, ob es Beschränkungen oder bestimmte Verpflichtungen gibt (z. B. Quellcodeoffenlegung) und ob die Lizenz Klauseln enthält, welche mit anderen verwendeten Lizenzen inkompatibel sind. Besonders beim Einsatz von OSS mit Copyleft-Klauseln (z. B. GPL oder AGPL) ist Vorsicht geboten, da hier abgeleitete Werke unter der gleichen OSS-Lizenz veröffentlicht werden müssen. Dies kann gegebenenfalls dazu führen, dass Geschäftsgeheimnisse offenbart werden.

Merke:

Open-Source-Software ist nicht unsicherer als proprietäre Software, erfordert jedoch ein angepasstes Verantwortungsmodell. Durch eine Kombination aus einer frühzeitigen, transparenten Dokumentation der Bedarfe und Anforderungen, dem Einsatz von externen Dienstleistern und einer fundierten Lizenzprüfung lassen sich Haftungsrisiken identifizieren und minimieren. Auf diese Weise können Verwaltungen OSS sicher und effizient einsetzen – ohne zusätzliche Haftungsrisiken für ihre Mitarbeitenden. Die Stadt München hat beispielsweise gezeigt, dass eine solche systematische Herangehensweise Unsicherheiten beim Einsatz von OSS abbauen kann.

1.3.4 Akzeptanzprobleme bei Mitarbeitenden

Vorbehalte gegenüber OSS können viele Ursachen haben. Ein Grund kann der Mangel an Vertrautheit und Erfahrung mit OSS sein, da im privaten Bereich typischerweise proprietäre Lösungen eingesetzt werden. Auch die offenen Gestaltungsmöglichkeiten von OSS können zu Unsicherheiten bei potenziellen Endanwendenden führen. Oftmals kann die Unkenntnis von entsprechenden Programmen im Bereich OSS als Alternative zu proprietärere Software kann ein Hindernis darstellen.

Zudem schüren Hersteller proprietärer Software gerne Vorurteile gegenüber OSS oder stellen ihre proprietären Lösungen als überlegen dar, um Geschäft zu generieren. Dabei werden die proprietären Lösungen oft als „professioneller“, „sicherer“ oder „performanter“ dargestellt und OSS Schwächen in diesen Bereichen unterstellt. Damit einher geht auch die Frage nach der Verantwortlichkeit für Themen wie die Sicherheit, welche bei proprietärer Software zumindest teilweise beim Hersteller gesehen wird, bei OSS jedoch in der eigenen Verantwortung liegt. All dies kann in Summe zu einer deutlich ablehnenden Haltung von OSS führen.

Dem kann nur durch Aufklärung entgegengewirkt werden. Hierfür sind zunächst entsprechende Informationsmöglichkeiten erforderlich, welche von gezielten Schulungen bis hin zu praxisnahen Anwendungsfällen (z. B. Demo-Nutzung von Anwendungen) reichen können, um die Vorbehalte

zu reduzieren oder ganz aufzulösen. Zudem kann eine behördeninterne Ansprechperson oder sogar eines Kompetenzteams für den Wissensaustausch aufgebaut werden, um Vorbehalte abzubauen und Akzeptanz zu fördern. Ebenfalls hilfreich ist es in diesem Zusammenhang, wenn Führungskräfte die Einführung von OSS aktiv unterstützen und diesbezüglich als Vorbild agieren.

Wenn es unter den Mitarbeitenden starke Vorbehalte gegenüber der Verwendung von OSS gibt, kann eine sofortige, flächendeckende Umstellung von Software Widerstände hervorrufen – insbesondere, wenn es sich um Software handelt, welche im täglichen Gebrauch ist. Eine solche ist deswegen nicht zu empfehlen. Stattdessen sollten Pilotprojekte mit einer kleinen Nutzergruppe gestartet werden, um erste Erfolge zu demonstrieren und frühzeitig Feedback zu sammeln. Dieses Feedback kann dann wiederum auch in die konkrete Ausgestaltung der OSS-Funktionalitäten führen: Da OSS oft Flexibilität in der Anpassung bietet, kann sie an spezifische Bedürfnisse der Mitarbeitenden angepasst werden, um die Benutzerfreundlichkeit zu verbessern.

1.3.5 Wissens- und Schulungsbedarf

Der Bedarf an Schulungen und Wissensmanagement im Bereich OSS ergibt sich aus der Vielfalt der Zielgruppen und Inhalte. Dabei ist es wichtig zu erkennen, dass das eigentliche Problem nicht in der Natur von OSS liegt, sondern in der geringen Nachfrage nach spezifischen Schulungen – ein Umstand, der nicht exklusiv für OSS gilt, sondern auch in anderen Nischenbereichen der IT zu beobachten ist.

Da OSS-Lösungen noch nicht flächendeckend im öffentlichen Sektor etabliert sind, gibt es im Vergleich zu proprietären Produkten weniger standardisierte Schulungsangebote. Das bedeutet, dass Verwaltungen gezielt eigene Schulungsstrategien entwickeln und gegebenenfalls individuelle Schulungsformate etablieren müssen.

Zielgruppen und Schulungsinhalte

- **Mitarbeitende in der Verwaltung:** Wie im Kapitel „Akzeptanzprobleme bei Mitarbeitenden“ erläutert, können Schulungen dazu beitragen, Vorbehalte gegenüber OSS abzubauen und die Nutzungskompetenz zu erhöhen. Schulungen für Endanwendende sollten sich auf die Anwendung und Handhabung der OSS-Gesamtlösung konzentrieren.

- **Entwickelnde und technische Fachkräfte:** Diese benötigen spezifische Schulungen zu Softwarearchitektur, relevanten Komponenten, Lizenzierungsmodellen sowie IT-Sicherheit und Datenschutz. Da OSS-Technologien oft eine hohe Entwicklungsdynamik aufweisen, ist eine kontinuierliche Weiterbildung essenziell.
- **Beschaffungsverantwortliche und Entscheidende:** Um OSS strategisch einzusetzen, sind Kenntnisse über OSS-Lizenzmodelle, Governance-Prinzipien und Vergabestrategien erforderlich.

Herausforderung: Geringe Nachfrage führt zu wenig Angebot

Da OSS in vielen Bereichen noch als spezielle Lösung betrachtet wird, existiert bislang ein begrenztes Angebot an standardisierten Schulungen. Anders als bei proprietärer Software, wo Hersteller Schulungen oft als Teil ihres Geschäftsmodells anbieten, müssen Schulungsangebote für OSS häufig individuell organisiert werden. Daher ist es notwendig, Schulungen nicht nur extern einzukaufen, sondern auch internes Wissen systematisch aufzubauen und weiterzugeben.

Community-Arbeit als Wissensquelle

Ein großer Vorteil von OSS ist die Möglichkeit, Wissen direkt aus der Community zu beziehen. Mitarbeitende, die an OSS-Projekten beteiligt sind, können ihr Wissen an Kolleginnen und Kollegen weitergeben. Da jedoch nicht alle Beschäftigten aktiv an Community-Prozessen teilnehmen können, sollten Gesundheitsämter überlegen, wie sich dieses Wissen effizient intern weitervermitteln lässt – sei es durch interne Workshops, Wissensplattformen oder Mentoring Programme.

Merke:

Die Herausforderung bei Schulungen zu OSS liegt nicht in der Software selbst, sondern in der bislang geringen Nachfrage, die zu einem begrenzten Schulungsangebot führt. Um diesen Engpass zu überwinden, sollten Verwaltungen frühzeitig in interne Wissensstrukturen investieren, Kooperationen mit der OSS-Community fördern und Schulungsbudgets für gezielte Weiterbildungen einplanen.

1.3.6 Gängige Open-Source-Lizenzen und ihre rechtlichen Implikationen

1.3.6.a Permissive Lizenzen (flexible Nutzung)

Hinweis:

Dieser Text dient der ersten Orientierung. Für konkrete Projekte oder Lizenzfragen sollte stets eine rechtliche Beratung durch Fachleute erfolgen.

MIT-Lizenz

Die MIT-Lizenz gehört zu den bekanntesten permissiven Open-Source-Lizenzen. Sie erlaubt die uneingeschränkte Nutzung, Modifikation und Weitergabe – auch in proprietärer Software. Die einzige Voraussetzung ist die Beibehaltung des ursprünglichen Copyright-Hinweises und Lizenztextes. Aufgrund ihrer Einfachheit ist sie mit nahezu allen anderen Lizenzmodellen kompatibel und besonders für Bibliotheken oder wiederverwendbare Komponenten attraktiv.

Apache 2.0

Die Apache-Lizenz ist ähnlich offen wie die MIT-Lizenz, enthält jedoch zusätzlich eine explizite Klausel zum Patentschutz. Diese gewährt den Nutzern Rechtssicherheit, falls Patentansprüche durch den Lizenzgeber bestehen. Sie ist mit vielen anderen Lizenzen kombinierbar und wird häufig in kommerziellen Projekten verwendet.

BSD-Lizenzen (2- und 3-Klausel)

Die BSD-Lizenzen zählen ebenfalls zu den permissiven Modellen. Die 2-Klausel-Version verlangt lediglich die Nennung des Urhebers und einen Haftungsausschluss. Die 3-Klausel-Version ergänzt eine sogenannte „No-Endorsement“-Klausel, die untersagt, den Namen der Urheber zu Werbezwecken zu verwenden. Beide Varianten lassen sich problemlos mit proprietärer Software kombinieren und sind in vielen technischen Open-Source-Projekten etabliert.

1.3.6.b Copyleft-Lizenzen (stärkere Verpflichtungen)

GPL v3 (GNU General Public License)

Die GPL v3 ist eine stark schützende Copyleft-Lizenz. Sie erlaubt freie Modifikation und Weitergabe, verlangt jedoch, dass abgeleitete Werke ebenfalls unter GPL veröffentlicht werden, sobald sie an Dritte weitergegeben werden. Dies kann für Unternehmen mit proprietären Lösungen problematisch sein, da der kombinierte Quellcode offengelegt werden müsste – insbesondere bei enger Kopplung oder statischem Linken.

AGPL v3 (Affero General Public License)

Die AGPL v3 erweitert die GPL v3 um eine zusätzliche Verpflichtung: Auch bei Bereitstellung als Webdienst (z. B. SaaS) muss der geänderte Quellcode offengelegt werden. Sie eignet sich besonders für Projekte, die maximale Transparenz und Nachvollziehbarkeit gewährleisten sollen, wird aber von kommerziellen Cloud-Anbietern oft gemieden.

EUPL (European Union Public Licence)

Die EUPL wurde speziell für europäische Behörden entwickelt. Sie ist eine Copyleft-Lizenz, die mit der GPL v2 sowie unter bestimmten Bedingungen einseitig mit der GPL v3 kompatibel ist – EUPL-Code darf in GPL-Projekte übernommen werden, umgekehrt ist dies nicht ohne Weiteres möglich. Für öffentliche Verwaltungen bietet sie einen passenden rechtlichen Rahmen mit europäischem Rechtsbezug.

1.3.6.c Praktische Herausforderungen und Lösungsansätze

1. Lizenzkonflikte vermeiden

Permissive Lizenzen (MIT, Apache 2.0, BSD) lassen sich gut in proprietäre Software integrieren. Copyleft-Lizenzen (GPL, AGPL) können bei enger Integration mit geschlossenem Code dazu führen, dass der gesamte verbundene Quellcode offengelegt werden muss.

Beispiel: Wird eine GPL-lizenzierte Bibliothek in ein proprietäres System statisch eingebunden, kann daraus die Pflicht zur Offenlegung des Gesamtprojekts entstehen.

2. Architektur und Modultrennung

Eine klare Trennung zwischen Open-Source- und proprietären Komponenten (z. B. via Microservices oder REST-APIs) hilft, Lizenzkonflikte zu vermeiden.

Gute Dokumentation und eindeutige Lizenzkennzeichnungen sind entscheidend für spätere Wartung und Compliance-Prüfungen.

3. Community vs. Unternehmensbedürfnisse

Permissive Lizenzen sind in der Regel wirtschaftsfreundlicher, da sie wenig Einschränkungen auferlegen.

Copyleft-Lizenzen stärken hingegen die kollaborative Weiterentwicklung und das Open-Source-Ökosystem.

Erfolgreiche Open-Source-Projekte wie Linux (GPL) oder Kubernetes (Apache 2.0) zeigen, dass unterschiedliche Lizenzmodelle je nach Zielsetzung sinnvoll eingesetzt werden können.

4. Fazit: Die richtige Lizenz wählen

Die Wahl der Lizenz sollte sich an den strategischen Zielen des Projekts orientieren:

- Maximale Flexibilität & kommerzielle Nachnutzung? MIT, Apache 2.0 oder BSD
- Gemeinschaftliche Weiterentwicklung & Schutz vor Kommerzialisierung? GPL oder AGPL
- Öffentliche Verwaltung in der EU? → EUPL

Bei Unsicherheit ist der Rückgriff auf rechtliche Expertise dringend zu empfehlen – insbesondere, wenn OSS in Kombination mit bestehenden proprietären Systemen oder im Rahmen von SaaS eingesetzt wird.

Tipp: Weitere Informationen und Lizenzvergleiche bietet z. B. https://choosealicense.com

1.4 Strategie zur Förderung von OSS im öffentlichen Sektor in Deutschland

Die Förderung von OSS im öffentlichen Sektor erfordert eine langfristige politische Strategie, die sowohl rechtliche Rahmenbedingungen anpasst als auch strukturelle Veränderungen ermöglicht. Ziel ist es, OSS nicht nur als alternative Option, sondern als den bevorzugten Standard in der Verwaltung zu etablieren. Um dies zu erreichen, müssen kurz-, mittel- und langfristige Maßnahmen ineinandergreifen.

Kurzfristige Maßnahmen – Transparenz und strukturelle Grundlagen schaffen

Ein erster Schritt ist die Schaffung von Transparenz über bereits existierende OSS-Lösungen in der öffentlichen Verwaltung. Ein öffentlich zugängliches Register für OSS im öffentlichen Sektor kann hier Abhilfe schaffen. Gleichzeitig müssen Behörden stärker für die Vorteile offener Software sensibilisiert werden. Pilotprojekte in verschiedenen Verwaltungsbereichen können zeigen, wie OSS erfolgreich eingesetzt werden kann, um Effizienz und digitale Souveränität zu steigern.

Ausschreibungen sollten so gestaltet werden, dass OSS nicht mehr strukturell benachteiligt wird. Bisherige Vergabeverfahren fokussieren oft auf kurzfristige Kosten und Herstellerabhängigkeit, anstatt den Total Cost of Ownership (TCO) oder Aspekte wie Interoperabilität, Flexibilität und Nachhaltigkeit zu berücksichtigen. Dabei wird OSS oft übersehen oder falsch eingeordnet: Da viele OSS-Lösungen keine Lizenzkosten haben, kann es Fälle geben, in denen eine Ausschreibung gar nicht notwendig ist.

Das Vergaberecht greift nämlich nur dann, wenn mit der OSS-Beschaffung zusätzliche entgeltliche Dienstleistungen verbunden sind – etwa für Anpassung, Wartung oder Support. Nach Einschätzung von ZenDiS bedeutet dies, dass bei einer kostenfreien OSS-Nutzung kein entgeltlicher Auftrag vorliegt und das Vergaberecht nicht anwendbar ist. Falls jedoch Dienstleistungen erforderlich sind, könnte es je nach Kostenhöhe möglich sein, diese direkt zu beauftragen, ohne eine förmliche Ausschreibung durchzuführen, solange die Vergabeschwellen nicht überschritten werden.

OSS muss dennoch als gleichwertige oder bevorzugte Option in öffentlichen Ausschreibungen verankert werden, insbesondere dort, wo Anpassung oder Support benötigt werden. Ein Problem in bisherigen Ausschreibungsverfahren ist, dass kurzfristige Kosten stärker gewichtet werden als langfristige Einsparungen, die durch OSS entstehen – etwa durch entfallende Lizenzgebühren oder bessere Interoperabilität. Um hier faire Wettbewerbsbedingungen zu schaffen, braucht es klare Richtlinien für Verwaltungen, die sicherstellen, dass OSS-Alternativen systematisch geprüft und bewertet werden.

Aktuelle Situation: In Deutschland existieren bereits erste Initiativen zur Förderung von OSS in der öffentlichen Verwaltung. Eine zentrale Plattform ist openCoDE, die als Austauschort für OSS innerhalb der öffentlichen Verwaltung dient. Diese Plattform ermöglicht es Behörden, gemeinsam Software zu entwickeln und zu nutzen, wodurch Synergien geschaffen und digitale Souveränität gestärkt werden. Dennoch bleibt die Herausforderung, OSS in der Praxis rechtssicher in bestehende Vergabeverfahren zu integrieren und die langfristigen wirtschaftlichen Vorteile zu berücksichtigen.

Mittelfristige Maßnahmen – Rechtliche Anpassungen und Marktentwicklung fördern

Nachdem erste Strukturen geschaffen worden sind, müssen mittelfristig die rechtlichen Rahmenbedingungen weiter angepasst werden. OSS sollte als förderwürdige Maßnahme in Digitalisierungsprogrammen des Bundes und der Länder verankert werden.

Eine zentrale Herausforderung besteht darin, die Fragmentierung in der Verwaltung zu reduzieren. Bisher entwickeln Kommunen, Länder und Bundesbehörden oft isolierte Softwarelösungen, ohne Synergien zu nutzen. Um dies zu ändern, müssen Plattformen für die gemeinsame Entwicklung und Nutzung von OSS geschaffen werden.

Aktuelle Situation: Es gibt bereits Bestrebungen, das Vergaberecht zugunsten von OSS anzupassen. Ein Gutachten empfiehlt, einen Vorrang für OSS in der Vergabeverordnung für öffentliche Aufträge (VgV) zu verankern. Einige Bundesländer wie Schleswig-Holstein, Thüringen und Bayern haben diesen Vorrang bereits gesetzlich festgeschrieben.

Langfristige Maßnahmen – OSS als Standard etablieren

Langfristig muss OSS als Standard für neue Softwareentwicklungsprojekte im öffentlichen Sektor verankert werden. Dies kann durch die Einführung eines OSS-Gesetzes erfolgen, das Verwaltungen dazu verpflichtet, bei Neuentwicklungen OSS-Alternativen zu bevorzugen.

Zusätzlich muss der Austausch mit europäischen Partnern intensiviert werden. Deutschland sollte sich stärker an EU-weiten OSS-Initiativen beteiligen, um gemeinsame europäische Standards für Verwaltungssoftware zu entwickeln.

Aktuelle Situation: Einige Bundesländer gehen bereits mit gutem Beispiel voran. Schleswig-Holstein hat eine „Open Innovation und Open Source Strategie“ verabschiedet, die den verstärkten Einsatz von OSS in der Verwaltung vorsieht. Auf Bundesebene wurde das Zentrum für Digitale Souveränität (ZenDiS) gegründet, das OSS-Vorhaben im öffentlichen Sektor bündelt.

1.5 Fazit

Die Nutzung von Open-Source-Software (OSS) bietet viele Chancen, erfordert aber auch einige Anpassungen, um diese optimal nutzen zu können. Wesentliche Erfolgsfaktoren sind die Verfügbarkeit eigener Ressourcen, geeignete rechtliche und organisatorische Rahmenbedingungen sowie die aktive Teilnahme an Communities.

OSS ist nicht grundsätzlich kostengünstiger oder technisch unsicherer als proprietäre Software. Es kommt auf den konkreten Anwendungsfall an. Die Freiheit der Gestaltung erlaubt es, Software insbesondere im Hinblick auf Sicherheit, Interoperabilität und Funktionalität an spezifische Anforderungen anzupassen. Diese Freiheit ist jedoch mit Verantwortung verbunden – etwa für Pflege, Weiterentwicklung und rechtliche Compliance.

Der größte Vorteil von OSS liegt in der bewussten Entscheidungsmöglichkeit und der daraus resultierenden Gestaltungsfreiheit. Das Stichwort hierzu lautet „digitale Souveränität“. Sie beschreibt die Fähigkeit, digitale Prozesse, Systeme und Infrastrukturen selbstbestimmt, unabhängig und sicher gestalten zu können. Digitale Souveränität erfordert mehr als Technologie: Sie umfasst organisatorische Befähigung, rechtliche Sicherheit, technische Kontrollierbarkeit und ein strategisches Betriebsmodell. Proprietäre Software kann – je nach Anbieter und Vertragsgestaltung – ebenfalls souverän betrieben werden, erfordert jedoch in vielen Bereichen ein erhöhtes Maß an Vertrauen gegenüber Dritten.

Ungünstige Abhängigkeiten von proprietärer Software lassen sich durch Open-Source-Ansätze nachhaltig reduzieren – sofern deren Einführung mit klarer Zielsetzung, geeigneter Governance und dem erforderlichen Know-how begleitet wird. Dennoch ist OSS nicht in jedem Fall die beste Lösung. In vielen Szenarien kann eine Kombination aus Open-Source- und proprietären Komponenten sinnvoll sein – etwa um vorhandene Standardsysteme mit offenen Schnittstellen zu erweitern oder hybride Betriebsmodelle zu etablieren.

Daher sollte die Entscheidung für oder gegen eine OSS-Lösung – ebenso wie für eine proprietäre oder gemischte Lösung – stets unter Berücksichtigung und Bewertung aller technischen, rechtlichen und strategischen Faktoren getroffen werden. Im besten Fall ist sie selbst bereits Ausdruck einer bewusst gewählten, „digital souveränen“ Haltung.

2 LEITFADEN

2.1 Grundlagen

Vor diesem Hintergrund sind bei der Durchführung von OSS-Projekten eine Reihe von Faktoren zu berücksichtigen. Diese mögen zum Teil trivial erscheinen, jedoch ist es von entscheidender Bedeutung, dass bei allen Beteiligten ein gemeinsames Verständnis und Klarheit über die Ziele und die Vorgehensweise besteht. Glauben alle Beteiligten nur, das gleiche Verständnis zu Zielen und Vorgehen zu haben, ohne dass dies gemeinsam abgestimmt und dokumentiert wurde, kann dies zu Missverständnissen und auseinanderlaufenden Zielen führen.

Der nachfolgende Leitfaden soll einen pragmatischen Ansatz liefern, in welchen Schritten bei Beschaffung und Entwicklung von OSS vorzugehen ist, was dabei zu beachten ist und erläutert Grundlagendefinitionen auf denen das Vorgehensmodell des Leitfadens beruht.

Grundlage hierfür ist vor allem die abgestimmte Bestimmung der Zieldefinition, des Vorgehens und der Ressourcen in folgenden Schritten.

2.1.1 Übersetzung der Anforderungen

In der Regel entsteht der Bedarf für eine neue Software aus einer fachlichen Anforderung heraus. Sie kann jedoch auch aus technischen Erfordernissen heraus entstehen – wenn beispielsweise eine bestimmte Software oder Version von Herstellern nicht mehr unterstützt wird. Aus der Identifikation der relevanten Anforderungen können wiederum technische Anforderungen in Richtung des hieraus entstehenden IT-Projektes definiert werden. Diese Anforderungen müssen, soweit bereits bekannt, entsprechend dokumentiert werden, damit auch hierüber Transparenz herrscht.

Da Fachbereich und IT in der Regel nicht immer die gleiche Sprache hinsichtlich Anforderungen und Erfordernissen sprechen, ist eine Person erforderlich, welche diese Übersetzung in beide Richtungen vornehmen kann. Das bedeutet, diese Person muss in der Lage sein, die fachlichen Anforderungen zu verstehen und daraus Schlüsse für die Anforderungen an die IT zu ziehen, zu konkretisieren und zu definieren. Dies kann für einen Projekterfolg entscheidend sein.

A. Übersetzende Person zwischen Fachbereich und IT definieren

Hierbei muss es sich um eine Person handeln, die sowohl die fachlichen Anforderungen verstehen kann, als auch hieraus entstehende erste Erfordernisse an die IT für sich identifizieren sowie mit der IT diskutieren kann. Dabei geht es um die Identifikation weiterer Anforderungen und die Erstellung eines daraus resultierenden Pflichten- / Lastenheftes.
Die hierbei gewonnenen Erkenntnisse können weiterhin Einfluss auf die Art der Realisierung der fachlichen Anforderungen haben.

2.1.2 Arten der Anforderungen

Bedarf aus fachlicher Anforderung	Wenn fachliche Anforderungen in Form von OSS abgebildet werden sollen, kann dies entweder in Form einer neuen Anwendung oder als Modifikation einer bestehenden Lösung erfolgen. In beiden Fällen ist auf Basis der fachlichen Anforderungen der erforderliche Lösungsansatz zu finden.
Bedarfe aus technischen Anforderungen	Sollen technische Anforderungen der Auslöser für ein entsprechendes Projekt sein, können diese durch interne oder auch externe Faktoren hervorgerufen werden (beispielsweise Kosten, Aufwand oder Herstellersupport). In diesem Fall orientiert sich die Lösung an der geeignetsten technologischen Lösung, welche gleichzeitig die fachlichen Anforderungen weiterhin sicherstellt.

Tabelle 4: Anforderungsvarianten

Für beide Anforderungsvarianten gilt: Änderungen bestehender Lösungen gelten als sogenannter „Change“ und liegen nicht im Fokus dieses Leitfadens.

Die Konsequenz aus der Art des Bedarfes liegt in der Verknüpfung von fachlicher Umsetzung und IT Projekt. Im Falle eines fachlichen Bedarfes ist in der Regel ein neues fachliches Verfahren zu erarbeiten, welches in der IT umgesetzt wird. Dabei arbeiten Fachbereich und IT bei der Erstellung des Verfahrens und der technischen Umsetzung Hand in Hand.

Hiervon unterscheidet sich ein Bedarf aus technischen Gründen. In diesem Fall ist bereits eine technische Lösung für ein fachliches Verfahren vorhanden, wobei die technische Lösung aus verschiedenen Gründen ersetzt werden muss. Neben der Übernahme des vorhandenen Verfahren in eine neue Lösung, kann dabei jedoch auch die Chance genutzt werden, um das fachliche Verfahren unter anderem durch technische Innovationen zu optimieren.

Aus den beiden unterschiedlichen Bedarfsansätzen kann ein unterschiedliches Vorgehen in der Realisierung resultieren.

Im Rahmen der Entwicklung der Lösung können sich beide Bereiche gegenseitig beeinflussen, um einen optimales Ergebnis zu erreichen (so kann die IT beispielsweise Prozessautomatisierungen vorschlagen, welche der Fachbereich noch manuell gesehen hat). Daher kann der Einsatz gemeinsamer Teams bei der Konzeption sinnvoll sein. Gemeinsame Teams tragen zudem zur Verbesserung des gegenseitigen Verständnisses bei.

B. Festlegung der Art der Anforderung

In den meisten Fällen wird diese aus fachlichen Bedarfen heraus begründet sein. Daher wird der Fokus auf der Erstellung neuer Anwendungen auf Basis fachlicher Anforderungen oder auf Basis des Ersatzes vorhandener Anwendungen liegen.

Die Festlegung der Art der Anforderung kann zudem Einfluss auf den Entwicklungsprozess haben, Schwerpunkte definieren (z. B. zeitliche Rahmenbedingungen) oder bestimmte personelle Ressourcen erfordern. Einer übersetzenden Person zwischen Fachbereich und IT zum gegenseitigen Verständnis der Anforderungen und Erfordernissen kommt dabei eine Schlüsselrolle zu.

2.1.3 Arten von OSS

OSS Anwendung	Oft handelt es sich um eine vorgefertigte fachliche Anwendung auf Basis von OSS. Hierbei kann es sich beispielsweise um eine fertige kommerzielle Lösung oder eine von der Community zur Nachnutzung bereitgestellte Lösung – möglicherweise auch in Form einer SaaS-Lösung – handeln. Anwendungen können auch das Ergebnis dedizierter eigener Entwicklungsarbeit, beispielsweise durch eigene Ressourcen oder durch beauftragte Dienstleister, sein. Für die Auswahl oder Entwicklung einer solchen Software oder Service-Lösung sind daher im Wesentlichen die fachlichen Anforderungen ausschlaggebend.
OSS Anwendungs-Komponenten und systemnahe Software	Hierbei handelt es sich um Anwendungskomponenten auf Basis von OSS, beispielsweise Java, Web-server oder Datenbanken, es können aber auch Low Code Entwicklungsframeworks oder Code-Zeilen (Snippets) sein. Diese Komponenten sind für sich allein normalerweise nicht in der Lage, fachliche Anforderungen abzudecken. Daher sind sie als Teile einer größeren Anwendung zu sehen. Eine Auswahl entsprechender Komponenten kann in der Regel im Rahmen von Entwicklungsarbeiten erfolgen, bei denen die jeweils geeignetsten Komponenten für den jeweiligen Zweck ausgewählt werden. Hierfür sind neben den fachlichen Anforderungen auch Anforderungen der IT, wie Betriebssicherheit und Support, Security, Bestand und Aktivität einer Community, Kosten und vieles mehr im Wechselspiel zu berücksichtigen. Zudem ist hierbei ein tiefes Verständnis im Bereich der Softwareentwicklung und Architekturen von OSS erforderlich. Auswahl und Verwendung solcher Komponenten können auch mit einem entsprechenden spezifischen Community Engagement einhergehen.
OSS-System-komponenten	Hierunter ist die maschinelle Ebene zu verstehen. Dies beginnt bei Betriebssystemen wie verschiedenen Linux-Derivaten, Laufzeit-Umgebungen oder Containerlösungen wie Docker, Virtualisierung oder Middleware. Solche OSS-Lösungen erfordern in der Regel ein tiefes Verständnis der Technologien und sind nicht durch fachliche Anforderungen bestimmt. Jedoch sind betriebliche Anforderungen und möglicherweise die Eignung für die Integration operativer Betriebsprozesse der IT zu berücksichtigen. Eine bessere oder schlechtere Eignung hierfür kann dauerhaft zu höheren oder geringeren Kosten führen. Auch hier können Auswahl und Verwendung entsprechender Systemkomponenten mit einem entsprechenden spezifischen Community Engagement einhergehen.

Tabelle 5: OSS-Arten

C. Klärung der erforderlichen Art der OSS

Da in erster Linie neue fachliche Anforderungen oder die Ablösung bestehender Anwendungen im Fokus stehen dürften, wird sich die Art der Software auf Anwendungen oder Anwendungskomponenten eingrenzen. Ob neben OSS-Anwendungen auch Anwendungskomponenten in dem Entscheidungsprozess betrachtet werden sollen oder ob diese ergebnissoffen dem IT-Projekt überlassen bleiben, kann ein Hinweis auf die potenziell erforderliche Form der Umsetzung / Beschaffung sein.

Systemkomponenten dürften von geringerer Relevanz bei der Auswahlentscheidung sein. Sie spielen von allem in der Schnittstelle zwischen Softwareentwicklung und IT-Betrieb eine Rolle und sind sekundär im Rahmen des Entwicklungsprozesses zu betrachten.

In allen Arten von OSS gilt, dass es diese Software auch als Teil proprietärer Lösungen geben kann, beispielsweise Laufzeitumgebungen oder Virtualisierungskomponenten, Java-Plug-Ins und Anwendungsdatenbanken bis hin zu ganzen Anwendungsteilen (sogenannten Modulen) für kommerzielle proprietäre Lösungen. Diese sind jedoch nicht im Fokus dieses Leitfadens.

2.1.4 Definition der eigenen Rolle hinsichtlich OSS

Wichtig ist die Beantwortung der Frage nach der eigenen zukünftigen Rolle in Bezug auf die OSS-Lösung.

Möchte man nur die Konsumentenrolle wahrnehmen und mit möglichst wenig Aufwand eine fachliche Lösung für den eigenen Bedarf haben oder möchte man sich in der Rolle des Innovationstreibers begeben?

Die Beantwortung dieser Frage kann deutlichen Einfluss auf die Auswahl der Art der Entwicklung und des Betriebs haben. Dies ist insbesondere dann der Fall, wenn die Teilnahme oder sogar das Initiieren einer eigenen Community erforderlich ist.

Daher kann der Wunsch (und der sich hieraus ergebende Anspruch) bezüglich der eigenen Rolle ein wesentlicher späterer Aufwandstreiber im Zusammenhang mit der OSS-Lösung darstellen und sollte wohl überlegt sein.

2.1.5 Arten von Rollen

Konsument ohne Community-beteiligung	OSS wird genutzt, aber eine explizite Beteiligung an technischen oder fachlichen Communities findet nicht statt. Dies ist vor allem dann möglich, wenn eine SaaS-Lösung oder eine kommerzielle Lösung durch Dienstleister vorliegt, bei welcher der Dienstleister die entsprechende Brücke zu Communities darstellt und diesbezügliche Aufgaben wahrnimmt.
Konsument mit passiver Community-beteiligung	Bei einer passiven Beteiligung an Communities werden Neuerungen und Informationen der fachlichen oder technischen Community aufgenommen, aber es findet selbst keine aktive Beteiligungen hinsichtlich Anforderungen, Entwicklungsarbeiten oder Diskussionen statt.
Konsument mit aktiver Community-beteiligung	Im Gegensatz zur passiven Communitybeteiligung wird in der aktiven Beteiligung an fachlichen oder technischen Diskussionen, der Anforderungsplanung und vielem mehr mitgewirkt. Community-Arbeit ist keine Einbahnstraße. OSS lebt von aktiver Beteiligung – und dazu gehört auch die Rückgabe von Verbesserungen, Bugfixes, Dokumentationen oder neuen Modulen. Diese Rückflüsse stärken nicht nur das Projekt selbst, sondern positionieren die öffentliche Hand auch als verlässliche Partnerin innerhalb digitaler Ökosysteme. Insbesondere im ÖGD-Kontext kann die Rückgabe von Wissen allerdings durch rechtliche, organisatorische oder datenschutzrechtliche Rahmenbedingungen eingeschränkt sein. Umso wichtiger ist es, bereits zu Beginn eines OSS-Projekts zu klären: • Welche Beiträge können und dürfen öffentlich zurückgespielt werden? • Gibt es interne Freigabeverfahren oder Richtlinien, die dies regeln? • Können Verbesserungen modular bereitgestellt werden (z. B. via Pull Requests)? Selbst kleinere Beiträge wie Fehlerberichte, Übersetzungen oder Hinweise zur Nutzbarkeit tragen zur Qualität von OSS bei – und fördern zugleich das Verständnis für quelloffene Entwicklungsprozesse in der Verwaltung.

Innovationstreiber mit aktiver Community-beteiligung	Als Innovationstreiber werden aktive eigene Überlegungen und Fortentwicklungsvorschläge in die Community eingebracht und auch in der Diskussion mit der Community priorisiert. Hierbei können auch eigene Prototypenentwicklungen stattfinden.
Innovationstreiber als Community-Owner (ggf. auch in Kombination als SaaS Anbieter) – Maintainer	Letztlich ist es auch möglich, auf Basis eigener Entwicklungen (beispielsweise selbst in OSS entwickelte Fachanwendungen) eine Community zu gründen und zu leiten. Dabei wird die Fachanwendung auch anderen potenziellen Nutzenden zugänglich gemacht (beispiels weise in Form einer SaaS-Lösung) und entsprechend der Bedarfe in Abstimmung mit der Community weiterentwickelt.

Tabelle 6: Rollen in OSS-Communities

D. Definition des eigenen Community-Engagements

Direkt proportional zum Engagement in der Community ist der eigene Aufwand. Die aktive Betreuung einer eigenen Community erfordert einen deutlich höheren Zeitaufwand als z. B. die Konsumentenrolle mit passiver Beteiligung. Daher sollte der Wunsch nach einer Rolle auch mit der Verfügbarkeit entsprechender Ressourcen einhergehen. Bei knappen Ressourcen sollte die eigene Wunschrolle gegebenenfalls überdacht werden.

2.1.6 Umsetzung und Beschaffung

Eigenentwicklung durch interne Ressourcen	Die Entwicklung der Anwendung erfolgt durch interne Ressourcen / Personen, sowohl fachliche als auch technische Aspekte betreffend. Hierzu sind entsprechend qualifizierte Ressourcen erforderlich, die in ausreichendem Umfang für das Projekt zur Verfügung stehen. Zudem ist dediziertes Know-how im Bereich OSS erforderlich. Dies betrifft sowohl Lizenzierung, Community als auch die spezifischen Entwicklungserfordernisse und Werkzeuge (beispielsweise Low Code Frameworks). Vorteile liegen vor allem in der vollständigen Kontrolle des Entwicklungsprozesses, der Umsetzung der fachlichen Anforderungen und dem Design der technischen und Anwendungsarchitektur.
Eigenentwicklung für intern durch Dienstleister	Die Entwicklung der Anwendung erfolgt durch beauftragte und qualifizierte Ressourcen (Dienstleister, Freelancer). Hierzu sind in der Regel finanzieller Mittel im ausreichendem Umfang erforderlich. Dabei kann unterschieden werden, ob es sich bei den dienstleistenden Ressourcen um Personen im internen Bereich handelt, also „externe Beschäftigte“, welche wie interne Ressourcen in die internen Entwicklungsabläufe integriert sind, oder ob es sich um die Vergabe eines Gewerkes an ein externes Unternehmen handelt. Dies ist insofern wichtig zu unterscheiden, da im Rahmen der Dienstleistung als Teil des internen Entwicklungsprozesses die Sicherstellung der Lizenzierung, Teilnahme an Community oder die Bereitstellung von Werkzeugen und Tools weiterhin in interner Verantwortung zu verorten ist. Bei der Beauftragung eines fixen externen Gewerkes kann die Bereitstellung von Werkzeugen als Teil des Entwicklungsprozesses in der Verantwortung des Dienstleisters liegen. Daher resultiert hieraus, je nach Abgrenzung ein unterschiedlicher interner Aufwand und Verantwortungsgrad. Zudem kann die Wahl dieser Variante vergaberechtliche Aspekte beinhalten, welche zu beachten sind. Insbesondere kann bei Nutzung externer Dienstleister eine Ausschreibung erforderlich sein.

Anpassung Nachnutzung durch interne Ressourcen	Bei der Nachnutzung einer vorgefertigten Lösung kann die individuell erforderliche Anpassung dieser Lösung in Form interner Eigenentwicklung stattfinden. Die Rahmenbedingungen sind vergleichbar mit dem vorhergehenden Punkt „Entwicklung durch interne Ressourcen“.
Anpassung Nachnutzung für intern durch Dienstleister	Die Nachnutzung kann auch durch einen externen Dienstleister, äquivalent der Eigenentwicklung für intern durch Dienstleister, erfolgen. Diese Variante kann ebenfalls vergaberechtliche Aspekte beinhalten, die betrachtet werden müssen. Insbesondere kann eine Ausschreibung erforderlich sein.
Anpassung Nachnutzung SaaS durch Dienstleister	Eine Abstufung der Anpassung Nachnutzung durch Dienstleister ist die Nutzung eines SaaS Services zur Abbildung der erforderlichen fachlichen Anforderungen. Da ein vorgefertigter SaaS-Service in der Regel die eigenen fachlichen Anforderungen nicht hundertprozentig abbildet (ggf. sind Anpassungen an Datenstrukturen und Schnittstellen erforderlich), sind auch hier zumeist individuelle Anpassungen nötig. Diese werden in der Regel durch den SaaS Dienstleister durchgeführt. Der Vorteil liegt hierbei darin, dass keine internen Ressourcen für die Anwendungsentwicklung benötigt werden. Erforderliche Anpassungen für Schnittstellen oder Datenmodelle können direkt von intern kommuniziert und durch den SaaS Dienstleister umgesetzt werden. In Bezug auf den SaaS-Dienstleister kann zudem zwischen einem privatwirtschaftlichen Dienstleister und einer anderen Stelle der öffentlichen Hand unterschieden werden. Dies hat insofern ggf. vergaberechtliche Auswirkungen, da einem privatwirtschaftlichen Dienstleister in jedem Fall eine Gewinnerzielungsabsicht sowie Marktwettbewerb unterstellt werden kann und gegebenenfalls eine Ausschreibung erfolgen muss. Die genauen Anforderungen sind im Einzelfall zu bewerten.
Anpassung kommerzieller Lösung durch Dienstleister	Hierbei handelt es sich im Gegensatz zur SaaS-Lösung weiterhin um eine lokale Anwendung (oder Hosting beim Dienstleister) einer vorgefertigten Lösung. Kommerziell ist hierbei in der Regel ein vereinbarter kostenpflichtiger Support (oder Betrieb). In diesem Zusammenhang erbringt der Dienstleister, vergleichbar zur SaaS-Lösung, die Anpassung der Lösung an den Bedarf. Dies kann Teil der Supportvereinbarung sein. Die Vorteile sind ebenfalls vergleichbar der SaaS-Lösung, wobei jedoch das Betriebselement weiterhin in der eigenen Verantwortung (Eigenbetrieb oder Beauftragung Hosting) liegt. Auch hier sind vergaberechtliche Aspekte zu beachten.

Tabelle 7: Entwicklungsarten

Abgeleitet aus Anforderungen und Definition der eigenen Rolle ergeben sich logische Formen der Umsetzung. Das bedeutet, wenn für sich die Rolle als Innovationstreiber definiert wurde, lässt sich dies in der Regel nicht mit der Entwicklungs- und Betriebsform einer SaaS-Lösung als Konsument vereinbaren (siehe untenstehend: Anpassung Nachnutzung SaaS durch Dienstleister / Wahrnehmung von Software as a Service (SaaS)-Diensten von Dienstleistern).

Eine Sonderform stellt die Umsetzung dar, bei der man selbst als SaaS- oder Lösungsanbieter auftritt. Hierbei ist zu beachten, dass man damit die Rolle eines Dienstleisters einnimmt – mit allen dazugehörenden Erfordernissen und Pflichten. Aufgrund der damit verbundenen Komplexität wird in diesem Leitfaden nicht weiter darauf eingegangen.

2.1.6.a Arten der Entwicklung

In allen Formen der Entwicklung sind bei Anpassungen der Software die lizenzrechtlichen Auswirkungen und Verpflichtungen zu beachten (beispielsweise Copyleft).

Die Form des nach der Entwicklung stattfindenden Betriebs lässt sich teilweise aus der Form der Entwicklung, beispielsweise aus der Nutzung einer SaaS-Lösung, ableiten. Andere sind mit

verschiedenen Formen der Entwicklung kombinierbar, sofern sich die jeweiligen Gewerke und Zuständigkeiten sauber, gegebenenfalls auch vertraglich, abgrenzen lassen. Dabei sind die Anforderungen unter Berücksichtigung etwaiger Mitwirkungspflichten an die vertraglichen Regelungen mit der Beschaffung und dem IT-Betrieb zu definieren.

Etwaige Ausschreibungs- und Vergabeverfahren sowohl die Entwicklung als auch den Betrieb betreffend sind immer dann relevant, wenn Dienstleister einbezogen werden. Deren Einbindung beziehungsweise Beauftragung kann je nach Verfahren eine gewisse Zeit in Anspruch nehmen.

Daher kann auch der Zeithorizont bis zur erforderlichen Umsetzung ein Indikator für ein bestimmtes Entwicklungsverfahren sein, da bei Eigenentwicklungen in der Regel die Zeit für Ausschreibung und Vergabe entfällt, somit bereits für die Entwicklung genutzt werden kann und daher die Gesamtentwicklungszeit potenziell verkürzt. Solche Überlegungen können in die Entwicklungsentscheidung einfließen.

2.1.6.b Arten des Betriebs (Betriebsmodelle)

Eigenbetrieb (fachlich / Anwen-dung / technisch)[1] intern durch eigene Ressourcen und Systeme	Sowohl der fachliche Betrieb, der technische Betrieb als auch der Anwendungsbetrieb erfolgen intern. Da der fachliche Betrieb immer intern stattfindet, ist zumeist eine Entscheidung hinsichtlich des technischen und des Anwendungsbetriebs zu treffen. Beim internen Betrieb erfolgt dieser im eigenen Rechenzentrum und auf eigenen technischen Plattformen (physische Server, Virtualisierung, Container, etc.). Hierzu ist neben den technischen Komponenten auch die Berücksichtigung der internen technischen Architekturen (von Serversystemen bis Netzwerk) und die Einbindung in die Betriebsprozesse erforderlich. Diese können je nach Anwendung und Software mehr oder weniger aufwändig sein. Daher kann bereits eine Optimierung der Lösung im Sinne eines einfachen Betriebs in die Entwicklung einfließen.
Eigenbetrieb (fachlich) und technischer Betrieb durch Dienstleister (Outsourcing / Hosting)	Neben dem eigenen technischen Betrieb ist der Betrieb durch einen Dienstleiter möglich (Hosting) und heute üblich. Der Anwendungsbetrieb kann hierbei entweder intern oder durch einen Dienstleister erfolgen. Ein Indikator hierfür kann die Art der Entwicklung sein: Bei einer vollständigen Eigenentwicklung ist die Vergabe des Anwendungsbetriebs an einen externen Dienstleister unwahrscheinlich. Neben den erforderlichen finanziellen Ressourcen sind auch vertragliche Grundlagen und Leistungsvereinbarungen sowie Schnittstellen prozessualer, technischer, fachlicher und organisatorischer Art zu regeln.
Wahrnehmung von SaaS-Diensten von Dienstleistern	SaaS Dienste sind in Bezug auf die erforderlichen Betriebsaufwände in der Regel am einfachsten zu handhaben. Allerdings spielt hierbei das Thema Kosten und Kostenkontrolle eine wesentliche Rolle, da die Kostenverrechnung in der Regel auf Basis der Nutzung stattfindet. Dabei kann zwischen zwei Varianten unterschieden werden: Zum einen eine SaaS-Lösung im öffentlichen Bereich. Die Lösung wird durch ein anderes Amt oder eine Behörde bereitgestellt. Hierbei kann im Bereich Kosten und Verrechnung Einfluss auf die besonderen Erfordernisse der öffentlichen Hand genommen werden. Zum anderen kann die SaaS-Lösung von einem Anbieter am Markt kommen. Hierbei ist der Einfluss auf Kostendynamik, -Begrenzung und Leistung entlang der Bedürfnisse im öffentlichen Gesundheitssektor deutlich geringer.

[1] **Fachlicher Betrieb:** Betreffend die fachlichen Inhalte und Prozesse;
Technischer Betrieb: Betreffend Rechenzentrum, Server, Netzwerke und Systemsoftware;
Anwendungsbetrieb: Betreffend den Betrieb der Anwendungssoftware, Webserver, Datenbanken etc.

Tabelle 8: Betriebsarten

E. Entscheidung zur Form der Entwicklung und des Betriebs

Bei der Wahl der Betriebsform sollten mehrere Faktoren berücksichtigt werden. Dabei spielen nicht nur direkte Kosten eine Rolle, sondern auch die möglichst medienbruchfreie prozessuale und organisatorische Integration in den eigenen Betrieb (sowohl fachlich, anwendungstechnisch als auch systemtechnisch). Hierdurch können die Qualität gesteigert und nachgelagerter Aufwand gegebenenfalls reduziert werden.

Eine wesentliche Rolle bei der Entscheidungsfindung werden auch die verfügbaren internen Ressourcen spielen. Für die Zeitplanung und damit die Entscheidungsfindung können auch Vergabeverfahren von Dienstleistern eine wesentliche Rolle spielen, da diese in der Regel einen gewissen Vorlauf benötigen, sofern nicht bereits entsprechende Rahmenvereinbarungen bestehen.

2.2 Ein Vorgehensmodell zur praktischen Entscheidungsfindung

Das Vorgehensmodell soll eine praktische Anwendungshilfe zur Entscheidungsfindung sein. Hierbei sollen bestimmte Fragestellungen helfen, die Kernpunkte zu identifizieren, welche für eine individuelle Entscheidungsfindung erforderlich sind.

Im Prinzip geht es dabei um lediglich drei Kernfragen:

- **Was brauche ich?**

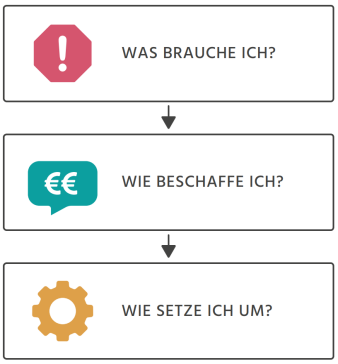
Ermittlung der fachlichen und technischen Anforderungen an die Softwarelösung.

- **Wie beschaffe ich?**

Festlegung der passenden Beschaffungsstrategie unter Berücksichtigung von OSS-spezifischen Vergabeverfahren.

- **Wie setze ich um?**

Planung und Durchführung der Implementierung, einschließlich Wartung und Weiterentwicklung.



Merke:

Die Fragen sollten gegebenenfalls intern mit Fachleuten diskutiert werden. Nicht immer ist eine auf den ersten Blick simple Fragestellung einfach zu beantworten.

2.2.1 Was brauche ich?

Was benötigt wird, ist grundsätzlich die erste Frage. Für die Entscheidung sind sowohl fachliche als auch technische Rahmenbedingungen ausschlaggebend.

- Konkretisierung der fachlichen Anforderung
- Klärung der technischen Rahmenbedingungen
- OSS als Lösungsansatz
- OSS-Lösung identifizieren und bewerten
- Software-Anforderungen konkretisieren
- Auswahlentscheidung

2.2.1.a Konkretisierung der fachlichen Anforderung

Die fachlichen Anforderungen bilden den Kern des Vorhabens und sind der entscheidende Faktor dafür, was benötigt wird. Beim Einsatz von OSS kommt zusätzlich die Frage nach Anpassbarkeit, Wartbarkeit und Sicherheit ins Spiel. Während OSS häufig flexible und kosteneffiziente Lösungen bietet, erfordert sie in der Regel eine sorgfältige Bewertung hinsichtlich Lizenzkonformität, langfristiger Wartung und möglicher Dienstleister-Unterstützung. Die Entscheidung sollte daher nicht nur auf funktionalen Anforderungen basieren, sondern auch Aspekte wie Community-Unterstützung, Sicherheitsupdates und Integrationsfähigkeit berücksichtigen.

Fragestellung: Was kann unsere Organisation in Entwicklung und Betrieb selbst leisten?

Prüffragen:

- ☐ Verfügen wir über ein OSPO?
- ☐ **Interne / personelle Ressourcen:**
  - ☐ Gibt es ausreichend qualifizierte Fachkräfte für Entwicklung, Betrieb und Support?
  - ☐ Sind diese im erforderlichen Maß verfügbar?
  - ☐ Können bei Bedarf zusätzliche Stellen für Fachkräfte geschaffen werden?
- ☐ **Finanzielle Ressourcen:** Sind Finanzmittel für eine potenzielle externe Unterstützung (Entwicklung, Support, Betrieb) im ausreichenden Maß vorhanden beziehungsweise beschaffbar?
- ☐ **Beschaffung / Rechtliche Vorgaben:**
  - ☐ Welche Beschaffungsrichtlinien, Vergaberegeln oder gesetzliche Vorgaben sind zu beachten?
  - ☐ Gibt es Rahmenvereinbarungen mit Dienstleistern oder Herstellern, welche den Bezug von Beschaf-fungen oder Dienstleistungen (Support, Entwicklung, Betrieb) vereinfachen?
- ☐ **Kooperationen:** Können andere Verwaltungen, Institutionen oder Gesundheitsämter eingebunden werden?

Ergebnis:

- ✓ **Bewertung der organisatorischen Machbarkeit sowie möglicher Unterstützungsstrukturen.**
- ✓ **Hinweise zu „Make“ or „Buy“ Ansatz**

2.2.1.b Klärung der technischen Rahmenbedingungen

Technische Rahmenbedingungen spielen eine entscheidende Rolle bei der Auswahl und Effizienz von IT-Lösungen, insbesondere bei OSS. Sie können die Auswahl einschränken, aber auch Synergien ermöglichen und Ressourcen schonen. Eine sorgfältige Analyse stellt sicher, dass die Software mit der bestehenden Infrastruktur kompatibel ist, Sicherheits- und Interoperabilitätsanforderungen erfüllt und langfristig skalierbar bleibt.

Fragestellung:

Welche technischen Voraussetzungen, Vorgaben beziehungsweise Rahmenbedingungen existieren?

Prüffragen:

- ☐ Gibt es vorhandene IT-Infrastrukturen, die berücksichtigt werden müssen?
- ☐ Welche Technologien (Betriebssysteme, Datenbanken, Cloud-Dienste) werden bereits genutzt?
- ☐ Gibt es technische Vorgaben (Sicherheit, Betrieb, Clients
- ☐ Sind Standards zur Interoperabilität notwendig (HL7, FHIR, etc.)?

Ergebnis: ✓ **Dokumentation von technischen Rahmenbedingungen und Restriktionen.**

2.2.1.c OSS als Lösungsansatz

Ob der Lösungsansatz zu den fachlichen Anforderungen auf Basis von OSS erfolgt, ist sowohl eine Frage der Funktionalität, der Effizienz, sowie gegebenenfalls der digitalen Souveränität. Daher kann die Analyse der Eignung von OSS auch zu einem negativen Ergebnis führen.

Fragestellung:

Ist OSS (besser) geeignet, die Anforderungen zu erfüllen?

Prüffragen:

1. Verfügbarkeit und Marktangebot

- ☐ Gibt es etablierte OSS-Lösungen, die unsere fachlichen Anforderungen erfüllen?
- ☐ Wird OSS bereits erfolgreich in anderen Behörden oder im öffentlichen Sektor genutzt?

2. Digitale Souveränität und Unabhängigkeit

- ☐ Trägt eine OSS-Lösung entscheidend zur Stärkung unserer digitalen Souveränität bei (z. B. durch Datenhoheit, Gestaltungsfreiheit, Unabhängigkeit von einzelnen Anbietern)?

3. Sicherheits- und Datenschutzkonformität

- ☐ Erfüllt die OSS-Lösung unsere spezifischen Sicherheits- und Datenschutzanforderungen (z. B. DSGVO, BSI-Vorgaben, NIS2)?
- ☐ Lässt sich die OSS-Lösung besser an unsere Sicher-heitsstandards anpassen als eine proprietäre Alternative?

4. Wirtschaftlichkeit und Effizienz

- ☐ Ist eine OSS-Lösung im Betrieb kostengünstiger und ressourcenschonender als eine proprietäre Lösung (z.B. durch Lizenzfreiheit, geringere Abhängigkeiten, Synergieeffekte mit anderen Behörden)?
- ☐ Ist die langfristige Wirtschaftlichkeit gewährleistet, insbesondere in Bezug auf Wartung und Weiterent-wicklung?

5. Community- und Ökosystem-Unterstützung

- ☐ Gibt es eine aktive OSS-Community oder relevante Projekte (z. B. im Gesundheitswesen, öffentlichen Sektor), die den Einsatz der Lösung erleichtern könnten?
- ☐ Ist eine Zusammenarbeit mit anderen Behörden oder Institutionen möglich, um Weiterentwick-lungen zu teilen?

6. Professioneller Support und Wartung

- ☐ Können für die OSS-Lösung professionelle Wartungs- und Supportdienstleistungen am Markt beschafft werden?
- ☐ Sind langfristiger Support und regelmäßige Sicherheitsupdates gesichert (z. B. durch Dienstleister oder eine aktive Community)?

7. Skalierbarkeit und Zukunftssicherheit

- ☐ Ist die OSS-Lösung zukunftssicher und kann sie langfristig weiterentwickelt und skaliert werden?
- ☐ Gibt es eine Roadmap für die Weiterentwicklung oder strategische Unterstützung durch öffentliche Institutionen?

Ergebnis: ✓ **Entscheidung über OSS als mögliche Lösung gegenüber klassisch proprietären Lösungen.**

2.2.1.d OSS-Lösungen identifizieren und bewerten

Sofern eine OSS-Lösung angestrebt wird, und es mehrere Lösungsansätze gibt sollte jene mit der besten Eignung unter Berücksichtigung aller Faktoren gewählt werden.

Fragestellung:

Welche OSS-Lösungen kommen in Frage?

Prüffragen – Identifizierung:

- ☐ Gibt es den Anforderungen entsprechende Lösungen im Markt (z. B. durch Recherchen auf Plattformen wie GitHub, OSOR, oder in Gesundheits-Communities)?
- ☐ Sind Erfahrungen anderer öffentlicher Verwaltungen zu OSS-Lösungsansätzen verfügbar?

Prüffragen – Bewertung:

- ☐ **Funktionalität:** Erfüllt die Software die Must-Have-Anforderungen?
- ☐ **Qualität:** Ist der Code stabil, dokumentiert und auditierbar?
- ☐ **Interoperabilität:** Kann die Software mit bestehenden Systemen integriert werden?
- ☐ **Community-Support:** Gibt es eine aktive und verlässliche OSS-Community?
- ☐ **Kosten:** Welche Kosten entstehen für Anpassung, Integration, Schulung und Wartung?
- ☐ **Lizenzierung:** Sind die Verpflichtungen aus der Lizenzierung transparent?
- ☐ **Betrieb und Ressourcen:** Wie aufwändig ist der Betrieb der Lösung (technisch, prozessual, personell)?

Ergebnis:

- ✓ **Eine Shortlist potenzieller OSS-Lösungen.**
- ✓ **Eine Vergleichsdarstellung potenzieller OSS-Lösungen.**

2.2.1.e Software-Anforderungen konkretisieren

Auf Basis einer Shortlist möglicher OSS-Lösungen sollte eine finale Auswahlentscheidung getroffen werden, welche sich an der Verfeinerung der Auswahlkriterien in Bezug auf die Anforderung orientiert.

Fragestellung:

Welche spezifischen Anforderungen sind für die finale Auswahl entscheidend?

Prüffragen:

- ☐ **Funktionale Anforderungen**
  (z. B. Benutzeroberfläche, Workflows)
- ☐ **Technische Anforderungen**
  (z. B. Hosting, Skalierbarkeit, Sicherheit)
- ☐ **Rechtliche Anforderungen**
  (z. B. Datenschutz, Lizenzmodelle)
- ☐ **Betriebliche Anforderungen**
  (z. B. Stabilität, Wartungsfreundlichkeit)
- ☐ **Wartung und Support:**
  - ☐ Wer kann dies langfristig sicherstellen?
  - ☐ Wartungsaufwand
  - ☐ Wartungskosten
- ☐ **Zukunftssicherheit:** Gibt es Potenzial für Weiterentwicklung?

Ergebnis:

- ✓ **Finale Anforderungsliste (Feinspezifikation) für die Auswahlentscheidung.**

2.2.1.f Auswahlentscheidung

Auf Basis der Feinspezifizierung und der Auswahl-Shortlist kann eine finale Auswahl für eine Lösung getroffen werden, welche die jeweiligen Anforderungen am besten trifft. Auf Basis dieser Auswahl kann dann eine Beschaffung beziehungsweise Entwicklung eingeleitet werden.

Fragestellung:

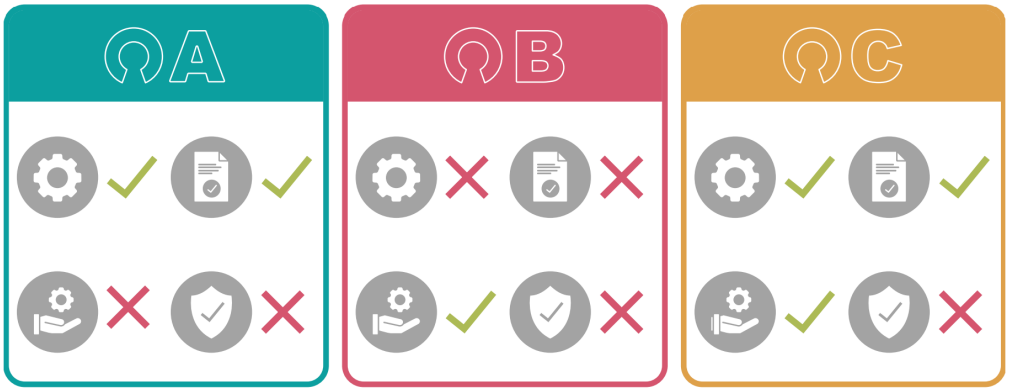
Welche OSS-Lösung erfüllt die Anforderung am besten?

Prüffragen:

- ☐ **Erfüllt die Software alle Must-Have-Anforderungen?**
- ☐ **TCO:** Wie hoch sind Anpassungs-, Implementierungs- und Betriebskosten?
- ☐ **Technische Machbarkeit:** Passt die Software in diebestehende IT-Landschaft?
- ☐ **Sicherheits- und Datenschutzprüfung:** Wurden alle Vorgaben erfüllt?
- ☐ **Langfristige Nachhaltigkeit:** Wie stabil ist die Community und das Projekt?

Ergebnis:

- ✓ **Auswahl der OSS-Lösung oder Entscheidung für Anpassung / Entwicklung.**



2.2.1.g Endergebnis

Der Prozess führt strukturiert von der fachlichen Anforderung über die Bewertung von OSS-Lösungen bis zur klaren Auswahlentscheidung. Dieses Vorgehen stellt sicher, dass OSS im öffentlichen Gesundheitswesen effizient und nachhaltig eingeführt wird.

2.2.2 Wie beschaffe ich?

Sofern noch keine Klarheit darüber besteht, wie eine OSS-Lösung potenziell entwickelt oder beschafft werden soll, empfehlen sich folgende erste Schritte, um eine fundierte Entscheidungsgrundlage zu erhalten:

- Vorgaben der organisatorischen Randbedingungen
- Umsetzungszeitrahmen
- Ziel der Innovationsführerschaft
- Empfehlung für ein Beschaffungsform / Entwicklungsform

2.2.2.a Vorgaben der organisatorischen Randbedingungen

Grundlage der Entwicklung und Beschaffung sind die organisatorischen Randbedingungen, welche Vorgaben schaffen, innerhalb deren sich Beschaffung und Entwicklung bewegen müssen. Dazu gehören verfügbare Ressourcen, rechtliche Vorgaben und die Frage, wie Wartung und Support sichergestellt werden können.

Fragestellung:

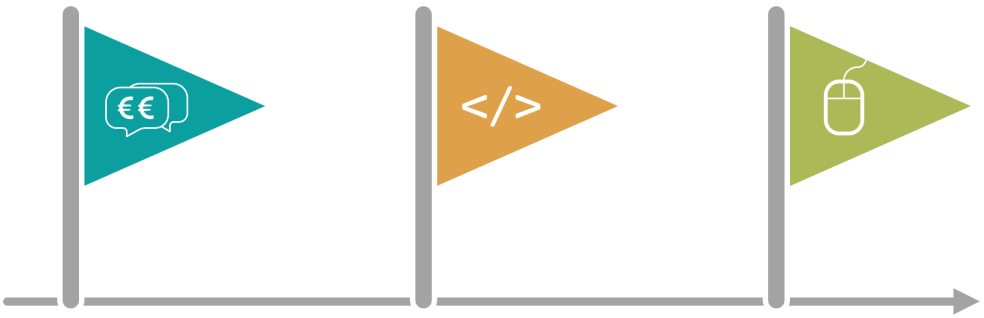
Was kann unsere Organisation in Entwicklung und Betrieb selbst leisten?

Prüffragen:

- ☐ Verfügen wir über ein OSPO?
- ☐ **Interne / personelle Ressourcen:**
  - ☐ Gibt es ausreichend qualifizierte Fachkräfte für Entwicklung, Betrieb und Support?
  - ☐ Sind diese im erforderlichen Maß verfügbar?
  - ☐ Können bei Bedarf zusätzliche Stellen für Fachkräfte geschaffen werden?
- ☐ **Finanzielle Ressourcen:** Sind Finanzmittel für eine potenzielle externe Unterstützung (Entwicklung, Support, Betrieb) im ausreichenden Maß vorhanden beziehungsweise beschaffbar?
- ☐ **Beschaffung / Rechtliche Vorgaben:**
- ☐ Welche Beschaffungsrichtlinien, Vergaberegeln oder gesetzliche Vorgaben sind zu beachten?
- ☐ Gibt es Rahmenvereinbarungen mit Dienstleistern oder Herstellern, welche den Bezug von Beschaffungen oder Dienstleistungen (Support, Entwicklung, Betrieb) vereinfachen?
- ☐ **Kooperationen:** Können andere Verwaltungen, Institutionen oder Gesundheitsämter eingebunden werden?

Ergebnis:

- ✓ **Bewertung der organisatorischen Machbarkeit sowie möglicher Unterstützungsstrukturen.**
- ✓ **Hinweise zu „Make“ or „Buy“ Ansatz**



2.2.2.b Umsetzungszeitrahmen

Der Zeitrahmen ist ein entscheidender Faktor für die Wahl der richtigen Vorgehensweise. Während eine bereits bestehende OSS-Lösung oft schneller implementiert werden kann, erfordert eine Eigenentwicklung in der Regel mehr Zeit. Des Weiteren sind je nach Vergabeform deutliche Vorläufe zu kalkulieren, insbesondere wenn es sich um umfangreiche Ausschreibungen handelt. Insofern können Zeitvorgaben ein kritischer Faktor werden. Zudem sollten in beiden Verfahren Zeitpuffer eingeplant werden, da Verzögerungen sowohl in Beschaffungsverfahren als auch bei Softwareentwicklung keine Seltenheit sind.

Fragestellung:

Welcher Zeitraum steht für Beschaffung oder Entwicklung zur Verfügung?

Prüffragen:

- ☐ **Zeitliche Dringlichkeit:** Gibt es zeitliche Vorgaben (gesetzliche Vorgaben – Beispiel Umsetzung einer gesetzlichen Regelung, externe Faktoren – Beispiel Produktabkündigung, interne Faktoren – Beispiel Personalverfügbarkeit) bis zur der eine Umsetzung / Einführung erfolgt sein muss?
- ☐ **Projektmeilensteine:**
  - ☐ Gibt es interne feste Fristen?
  - ☐ Gibt es externe Abhängigkeiten?
  - ☐ Gibt es abhängige Vertragslaufzeiten / Vertragsfristen?
- ☐ **Beschaffung / Rechtliche Vorgaben:**
  - ☐ Ist ein Pilotbetrieb möglich, um Risiken zu minimieren?
  - ☐ Sind zeitliche Puffer einkalkuliert?

Ergebnis:

- ✓ **Realistische Zeitplanung mit definierten Meilensteinen und Umsetzungsoptionen.**
- ✓ **Transparenz über zeitliche Vorgaben und Rahmenbedingungen sowie Abhängigkeiten.**

2.2.2.c Ziel der Innovationsführerschaft

OSS kann nicht nur bestehende Prozesse optimieren, sondern auch als Innovationsmotor dienen. Insbesondere die Gestaltungsfreiheit, welche mit OSS-Lösungen einhergeht, ermöglicht es, Lösungen optimal an Bedürfnisse anzupassen, neue Lösungswege zu denken und Lösungen für Bedarfe zu realisieren, für die es bisher keine proprietären Ansätze gibt.

Der strategische Einsatz von OSS kann zudem zur technologischen Unabhängigkeit beitragen und die digitale Souveränität stärken.

Fragestellung:

Bietet OSS Vorteile gegenüber proprietärer Software?

Prüffragen:

- ☐ Können durch OSS-Lösungen im Vergleich zu proprietärer Software die Anforderungen besser umgesetzt werden?
- ☐ Kann die digitale Souveränität, Hersteller-unabhängigkeit, technische Unabhängigkeit, Datenhoheit und IT-Sicherheit durch eine OSS-Lösung gestärkt werden?
- ☐ Können durch Kooperationen mit anderen OSS nutzenden Ämtern Synergien erreicht werden?
- ☐ Können durch OSS-Lösungen politische Initiativen unterstützt (siehe Gaia X), und gegebenenfalls Fördermittel eingeworben werden?

Ergebnis:

- ✓ **Einordnung von OSS in die IT-Strategie der Ämter**

2.2.2.d Empfehlung für eine Beschaffungs- oder Entwicklungsform

Aus den vorgenannten Schritten können Indikatoren für geeignete Beschaffungs- oder Entwicklungsformen unter den gegebenen Rahmenbedingungen abgeleitet werden. Natürlich können auch abseits dieser Empfehlung heraus Entscheidungen getroffen werden, jedoch sollten diese bewusst getroffen werden.

Fragestellung:

Wie / durch wen, soll eine OSS-Lösung entwickelt werden? Wie soll beschafft werden?

Prüffragen:

- ☐ Kann eine eventuelle Eigenentwicklung Ressourcen- (personell, Know-how, Geldmittel) und zeittechnisch geleistet werden?
- ☐ Ist externe Unterstützung (Freelancer, Entwicklung durch Dienstleister) erforderlich?
- ☐ Soll eine fertige Lösung vom Markt beschafft werden?
- ☐ Soll eine Lösung aus dem Behördenumfeld bezogen werden?
- ☐ Kann ein bestehender Rahmenvertrag oder Kooperationsvertrag genutzt werden (Entwicklung, Beschaffung, Support)?
- ☐ Muss ein Ausschreibungsverfahren gewählt werden?
- ☐ Welche Form der Ausschreibung muss gewählt werden?

Ergebnis:

- ✓ **Fundierte Empfehlung für die beste Umsetzungs- / Beschaffungsform unter Berücksichtigung wirtschaftlicher, technischer, organisatorischer und zeitlicher Aspekte.**

2.2.2.e Endergebnis

Ein strukturierter Auswahlprozess zur Entscheidungsfindung, wie eine OSS-Lösung entwickelt werden soll, ob externe Unterstützung erforderlich ist oder ob direkt eine Lösung vom Markt oder anderen Behörden bezogen werden soll, kann sowohl Kosten, Qualität als auch Umsetzungszeit optimieren.

2.2.3 Wie setze ich um?

Auch wenn zwar klar ist, wie beschafft beziehungsweise entwickelt werden soll, ist damit noch nicht geregelt, was im Rahmen der Umsetzung alles zu beachten ist. Auch hierbei spielen viele Aspekte eine Rolle, welche in den nachfolgenden Schritten umrissen werden.

2.2.3.a Fachliche Anforderung festlegen

Grundlage der Entwicklung beziehungsweise Einführung sind die fachlichen Anforderungen. Sie können sowohl die zeitlichen Rahmenbedingungen, die Sicherheit und den Datenschutz sowie die Gestaltung der Softwarelösung im Rahmen der fachlichen Prozesse, der Schnittstellen und der erforderlichen Performance der Systeme beeinflussen.

Fragestellung:

Was ist der konkrete Bedarf und welche Anforderungen gibt es?

Prüffragen:

Unter anderem Anforderungen bezüglich folgender Aspekte:

- ☐ Welche zeitlichen Vorgaben bis zur Produktivsetzung der Lösung gibt es?
- ☐ Welche fachlichen Prozesse müssen in der Lösung umgesetzt werden?
- ☐ Sind die fachlichen Prozesse bereits beschrieben?
- ☐ Welche Anforderungen hinsichtlich IT-Sicherheit und Datenschutz müssen umgesetzt werden?
- ☐ Gibt es regulatorische / gesetzliche Anforderungen denen die Lösung genügen muss?
- ☐ Gibt es Schnittstellen zu anderen technischen Systemen, welche Vorgaben / Anforderungen an die Lösung stellen?
- ☐ Gibt Vorgängersysteme deren Funktionalitäten oder Daten übernommen werden sollen?
- ☐ Gibt es fachliche Anforderungen an die Datensicherung beziehungsweise -bevorratung?
- ☐ Gibt es Anforderungen hinsichtlich der Performance?
- ☐ Muss die Lösung barrierefrei sein?
- ☐ Welche Anforderungen sind „muss“, welche sind „kann“?

Ergebnis:

- ✓ **Priorisierte Auflistung der bei der technischen Entwicklung der OSS-Lösung zu beachtende fachliche Anforderungen.**

2.2.3.b Technische Anforderungen festlegen

Bedingt durch die fachlichen Anforderungen, die betrieblichen Anforderungen und die verwendete Software entstehen Anforderungen an die Technologie. Sie schränken den Entwicklungsspielraum gegebenenfalls ein und müssen bei der Entwicklung der technischen OSS-Lösung beachtet. Dadurch sollen zusätzliche Komplexität, Fehlerquellen oder Kosten vermieden sowie unter anderem Betriebssicherheit, IT-Sicherheit, Datenschutz und Qualität gewährleistet werden.

Fragestellung:

Was sind die konkreten technischen Anforderungen?

Prüffragen:

Unter anderem Anforderungen bezüglich folgender Aspekte:

- ☐ IT-Sicherheit und Datenschutz (beispielsweise Verschlüsselung, DMZ, Zertifikate).
- ☐ Datensicherung und -bevorratung (beispielsweise Worm-Medien, Widerherstellungszeiten).
- ☐ Ausfallsicherheit (beispielsweise Virtualisierung, Containerisierung, Georedundanz).
- ☐ Performance (Systemanforderungen, Speicher, Prozessorleistung, Software- / Datenbankperformance, Transaktionsvolumen).
- ☐ Schnittstellenanforderungen, (offene) Schnittstellenstandards.
- ☐ Datenmodelle- / Vorgaben an Datenmodelle durch Dritte / andere Anwendungen.
- ☐ Wo erfolgt der technische Betrieb (beispielsweise Cloud, virtuelle Serverfarm, eigenes RZ, Hosting beim Dienstleister)?

Ergebnis:

- ✓ **Liste erforderlicher technischer Rahmenparameter aus fachlichen Anforderungen.**

2.2.3.c Organisatorische Anforderungen festlegen

Mit der Einführung und den Betrieb einer OSS-Lösung gehen auch eine Reihe organisatorischer Anforderungen einher. Manche dieser Anforderungen sind äquivalent zur Einsatz und Betrieb proprietärer Software zu sehen, machen sind OSS-spezifisch. Die Kenntnis dieser Anforderungen ist vor allem für die betriebliche Vorbereitung im Rahmen der Implementierung notwendig. Sie spielt jedoch auch eine Rolle bei der Entwicklung der Lösung, da die Kenntnis dieser Anforderungen eine Optimierung der Software in Richtung Aufwandsarmut im organisatorischen Bereich ermöglicht und somit Ressourcen schont.

Fragestellung:

Was sind die konkreten organisatorischen Anforderungen?

Prüffragen:

Unter anderem Anforderungen bezüglich folgender Aspekte:

- ☐ Gesamt-, Betriebsverantwortung und -Prozesse (fachlich, anwendungstechnisch, systemtechnisch).
- ☐ Organisatorische Zuordnung.
- ☐ Schnittstellenabgrenzung (unter anderem Verantwortung, Zuständigkeiten – technisch, prozessual, fachlich) – intern, zu Dienstleistern, Herstellern, Schnittstellensystemen und Schnittstellenfachprozessen.
- ☐ Einhaltung von Compliance-Anforderungen (unter anderem Lizenzierung, IT-Sicherheit, Datenschutz, Risikomanagement).
- ☐ Mittelverwaltung (unter anderem Budget- und Haushaltsmittelplanung, -Verwaltung, -Kontrolle, Stellenplanung und -Verwaltung).
- ☐ Berichtswesen (unter anderem Controlling, Risikomanagement, Compliance, Dienstleistersteuerung).

Ergebnis:

- ✓ **Liste erforderlicher organisatorischer Rahmenparameter aus fachlichen Anforderungen.**

2.2.3.d Methodische Herangehensweise festlegen und Zeitplanung

Sind die Anforderungen detailliert, kann geklärt werden, wie das Projekt methodisch umgesetzt werden kann. Hierbei können verschiedene Ansätze mit ihren spezifischen Vor- und Nachteilen für das Projekt tauglich sein. Daher kann die Wahl der Umsetzungsmethodik einen direkten Einfluss auf die Realisierung haben. Zudem ist es erforderlich, dass die Beteiligten in der jeweils besten Methodik geschult sind. Insbesondere in der Zusammenarbeit mit Externen und Dienstleistern ist zudem zu klären, inwieweit die Methodik hier abzugrenzen beziehungsweise die Zusammenarbeit zu regeln ist.

Ein wesentlicher Punkt ist zudem die Festlegung der Methode der Vergabe. Diese spielt eine wesentliche Rolle bei der Zeitplanung. Daher ist in der Folge der Definition der Vergabemethode eine Ableitung wesentlicher zeitlicher Eckparameter für eine anschließende genaue Projektplanung und Vorbereitung erforderlich.

Fragestellung:

Welche Entwicklungs- / Beschaffungsmethodik ist am besten geeignet und wird verwendet?

Prüffragen:

- ☐ Welche Projektmanagement Methoden werden verwendet (Agil: Beispiel Scrum / Kanban, klassisches Projektmanagement: Beispiel Prince2)?
- ☐ Welche Werkzeuge und Methoden werden zur Projektsteuerung verwendet?
- ☐ Werde entsprechende Qualifikationen oder Zertifikate bei Lieferanten und Dienstleistern gefordert?
- ☐ Wie gestaltet sich die Zusammenarbeit mit Lieferanten und Dienstleistern?
- ☐ Welche Berichtswege und Formen sind erforderlich (intern, mit Dienstleistern)
- ☐ Welche Kriterien sind bei einer Vergabe / Ausschreibung zu beachten?
- ☐ Welche Vergabemethode ist erforderlich?
- ☐ Welcher Vergabevorlauf resultiert aus der Vergabe-methode?
- ☐ Wie sieht der Zeitplan mit realistischen Meilensteinen aus (unter anderem Vergabevorbereitung, Vergabe, Projektinitialisierung, Entwicklung (inkl. Test und ggf. Pilotierung), Betriebseinführung und Übergabe)?

Ergebnis:

- ✓ **Verwendete Projektmethode im Kontext mit Dienstleistern.**
- ✓ **Vergabeverfahren.**
- ✓ **Projektplanung entlang wichtiger Meilensteine.**

2.2.3.e Projekt vorbereiten

Die Vorbereitung und Planung des Projektes schließt sowohl die Bereitstellung der erforderlichen Mittel und Ressourcen, die Vergabe und die Projektinitialisierung bis zur eigentlichen Umsetzung ein. In ihr sind eine Reihe von Tätigkeiten enthalten, welche unter anderem aus den individuellen Anforderungen der jeweiligen Organisation resultieren, welche das Projekt umsetzen möchte.

Fragestellung:

Was sind die spezifischen Projektvorbereitungserfordernisse?

Prüffragen:

- ☐ Liegen alle Kosten- und Mittelbedarfe (TCO) sowie die Anmeldung der Bedarfe vor?
- ☐ Liegen alle erforderlichen Gremienanmeldungen und Freigaben vor?
- ☐ Liegen Mittel- und Ressourcenfreigaben vor?
- ☐ Liegen alle erforderlichen Spezifikationen vor?
- ☐ Sind die erforderlichen internen Personalressourcen vorhanden und verfügbar?
- ☐ Gibt es Pilot- oder Machbarkeitsstudien, um Risiken zu minimieren?
- ☐ Ist die Vergabe eingeleitet / erfolgt?
- ☐ Liegen Projektplanung und Methodik vor?
- ☐ Ist ein Startdatum festgelegt?
- ☐ Wurden Qualitätskriterien und deren Messung definiert?
- ☐ Sind Risiken transparent und wurden Maßnahmen zur Risikoreduktion getroffen?

Ergebnis:

- ✓ **Das Projekt wurde vorbereitet und kann zu einem definierten Termin umgesetzt werden.**

2.2.3.f Projektdurchführung und Betriebsübergabe

Aufgrund der Flexibilität von OSS in Gestaltung und Nutzung ist eine nachhaltige Kontrolle zur Erreichung der fachlich-technischen Ziele erforderlich. Alle Entwicklungen sollten unter dem Gesichtspunkt der späteren aufwandsarmen betrieblichen Nutzung erfolgen.

Zudem sollten die späteren betrieblichen Aspekte in der Partizipierung an Community-Aktivitäten eingebracht werden. Dies sollte zudem auch in der betrieblichen Integration und Übergabe bedacht werden.

Sodann sind gerade durch das hohe Maß an Eigenverantwortlichkeit bei OSS die Themen IT-Sicherheit und Datenschutz zentrale Elemente. Das bedeutet, die Entwicklung sollte mit einem besonderen Fokus auf diese Aspekte stattfinden und kontrolliert werden (gegebenenfalls unter Einbindung der IT-Sicherheit und des Datenschutzes).

Im Sinne der betrieblichen Nachhaltigkeit und sofern OSS umfassendere Verwendung finden soll, sollte über die Einrichtung eines OSPO nachgedacht werden.

Fragestellung:

Wann ist das Projekt erfolgreich durchgeführt und in den Betrieb übergeben?

Prüffragen:

- ☐ Wurden die fachlichen Ziele und Anforderungen erfüllt?
- ☐ Wurde das Projekt im Zeit und Budgetrahmen umgesetzt?
- ☐ Findet eine Community Beteiligung entsprechend eigenen Rollenanforderungen statt?
- ☐ Sind die Anforderungen für einen aufwandsarmen und den Prozessen entsprechenden Betrieb umgesetzt?
- ☐ Ist die Software „sicher“ entsprechend Datenschutz und IT-Sicherheitsanforderungen?
- ☐ Ist die Betriebseinführung erfolgreich durchgeführt?
- ☐ Hat eine erfolgreiche Betriebsübergabe stattgefunden?

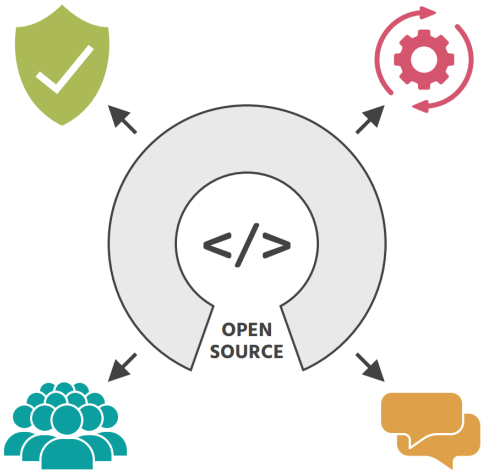
Ergebnis:

- ✓ **Das Projekt wird erfolgreich durchgeführt und das Endprodukt in den Betrieb übergeben.**

2.2.3.g Sicherstellung der Pflege der Software und kontinuierliche Verbesserung

Gerade durch die dauerhafte Dynamik bei der Weiterentwicklung von OSS im Rahmen der Communities ist eine nachhaltige Sicherstellung der Pflege und Aktualisierung der Lösung erforderlich. Dies gilt insbesondere unter den genannten Sicherheits- und Datenschutzaspekten.

Auch können je nach Rolle in der Community-Beteiligung eigene Anforderungen bei der Weiterentwicklung eingebracht werden. Außerdem ist nachhaltig die Sicherstellung der fachlichen Weiterentwicklung entsprechend der Bedarfe zu gewährleisten.



Fragestellung:

Wie intensiv müssen Pflege und Weiterentwicklung betrieben werden?

Prüffragen:

- ☐ Gibt es Prozesse (beispielsweise Anforderungsmanagement) zur regelmäßigen Weiterentwicklung der Software?
- ☐ Gibt es ein definiertes Budget zur Weiterentwicklung und Pflege unter Einbeziehung von Community Aktivitäten?
- ☐ Wer übernimmt Software-Wartung, Support und die Umsetzung neuer Anforderungen?
- ☐ Wie werden Bugs und Änderungswünsche dokumentiert und priorisiert?
- ☐ Wie wird an der Community (an welcher – Fachliche, Software, Lizenz-Community) teilgenommen?
- ☐ Wie bleibt die Software sicher und zukunftsfähig?
- ☐ Wird die IT-Sicherheit regelmäßig geprüft?
- ☐ Werden Schnittstellen und Datenmodelle übergreifend abgestimmt und gepflegt?
- ☐ Wird die Software im Sinne der betrieblichen Nachhaltigkeit und Aufwandsreduktion optimiert?

Ergebnis:

- ✓ **Konzept zur nachhaltigen Sicherstellung der Anforderungen, IT-Sicherheit und betrieblichen Optimierung**

2.2.4 Zusammenfassung: Ein strukturierter Leitfaden zur Umsetzung

Der strukturierte Ablauf von der fachlichen Anforderung über die technische Planung, die Umsetzung und die langfristige Weiterentwicklung ergibt einen umfassenden Leitfaden zur erfolgreichen Softwareentwicklung oder -beschaffung. Dieser Ansatz soll ein Vorgehensmodell darstellen, in dem alle relevanten Fragestellungen bedacht, adressiert und nachhaltig gelöst werden. Jenseits dessen können (und müssen) zusätzliche Fragestellungen und Priorisierung individuell einfließen.

2.3 Nachnutzung von OSS-Lösungen im Rahmen von SaaS

Die Nachnutzung von Open-Source-Software im Rahmen von Software-as-a-Service (SaaS) bietet öffentlichen Einrichtungen die Möglichkeit, leistungsfähige Anwendungen mit reduziertem Eigenaufwand zu betreiben. Dies kann besonders dann sinnvoll sein, wenn interne Entwicklungs- und Betriebskapazitäten fehlen. Dennoch ist SaaS – unabhängig vom Lizenztyp – nicht automatisch der einfachere oder schnellere Weg.

Der Erfolg einer SaaS-Implementierung hängt maßgeblich von der Standardisierung der zugrundeliegenden Prozesse ab. In der Praxis erfordert dies nicht nur eine geeignete technische Plattform, sondern auch organisatorische und strategische Abstimmungen. Die Konsolidierung und Harmonisierung von Verwaltungsprozessen ist ein komplexer und zeitintensiver Vorgang. Software kann diese Entwicklungen unterstützen – sie kann sie jedoch nicht ersetzen.

Darüber hinaus gilt: Auch proprietäre Standardsoftware kann in Form von SaaS betrieben werden und ermöglicht ebenfalls Nachnutzungseffekte – z. B. durch zentrale Pflege, standardisierte Schnittstellen und skalierbare Infrastruktur. Ein direkter Vergleich mit OSS-SaaS-Modellen ist daher nur im jeweiligen Anwendungskontext sinnvoll.

Die Wahl des SaaS-Modells – ob auf Basis von OSS oder proprietärer Lösungen – sollte unter Berücksichtigung der digitalen Souveränität getroffen werden. Entscheidend ist dabei weniger die Lizenzform als vielmehr:

- Wer betreibt die Infrastruktur?
- Wo werden die Daten gespeichert?
- Wer kann Updates kontrollieren, Zugriffsrechte verwalten oder Dienste erweitern?

2.3.1 SaaS im Kontext der aktuellen Herausforderungen der Gesundheitsämter

Die Gesundheitsämter stehen aktuell vor einer Reihe von Herausforderungen, bei denen die Nachnutzung von OSS im Rahmen von SaaS einen potenziellen Lösungsansatz darstellen kann.

Wesentliche Herausforderungen, denen sich die Gesundheitsämter stellen müssen, sind hierbei unter anderem:

- **Steigender Bedarf:** Die COVID-19-Pandemie hat gezeigt, wie wichtig effiziente Softwarelösungen unter anderem für die Nachverfolgung von Infektionen und die Koordination von Maßnahmen sind. Daher steigt der Bedarf an adäquaten modernen Lösungen bei den Gesundheitsämtern. Hierbei sind oft komplexe Sachverhalte und große Datenmengen zu verarbeiten, was die vorhandenen IT Systeme vor starke Herausforderungen stellt.
- **Begrenzte Ressourcen:** Als Organisationen der öffentlichen Hand sind auch die Gesundheitsämter mit zunehmend knappen Haushaltmitteln und Ressourcen konfrontiert. Daher kann sowohl der Ausbau als auch die Umsetzung neuer Anforderungen im Gesundheitswesen allein aus haushalterischer Sicht schnell an ihre Grenzen stoßen. Auch sind die begrenzten technischen und personellen Kapazitäten im öffentlichen Bereich ein Faktor, der betrachtet werden muss.
- **IT-Sicherheits- und Datenschutzanforderungen:** Insbesondere im sensiblen Gesundheits-bereich sind die IT-Sicherheits- und Datenschutzanforderungen besonders hoch. Hierbei spielt nicht nur die Einhaltung der DSGVO bei der Verarbeitung sensibler Gesundheitsdaten eine Rolle, sondern auch das Erfordernis, die Hoheit über die eigenen Daten zu behalten. Aus die-sen Gründen könnte sich die Nutzung von Cloud-Lösungen kommerzieller Marktanbieter schwierig gestalten, insbesondere, wenn diese nicht den Zugriff auf Daten aus dem außer-europäischen Ausland heraus ausschließen.

2.3.2 Wertbeiträge zur Bewältigung der Herausforderungen durch SaaS

Hieraus resultieren die Anforderungen nach Kostensynergien sowie Ressourcen und Kapazitätsflexibilität. SaaS zahlt auf diese Ziele in folgender Form ein:

- **Kosteneffizienz und -flexibilität:** SaaS Modelle reduzieren Investitionen in eigene Infrastruktur und können die laufenden Kosten durch geteilte Ressourcen reduzieren. SaaS bedeutet jedoch nicht automatisch Kostenreduktion. Die Kosten in SaaS orientieren sich in erster Linie an der Nutzung. Das bedeutet, dass bei geringer Nutzung auch geringe Kosten anfallen – bei umfangreicherer Nutzung allerdings höhere Kosten. Diese Flexibilität erfordert eine verstärkte Nutzungs- und Kostenkontrolle, um vorhandene Haushaltmittel nicht zu überschreiten.
- **Skalierbarkeit:** Der Grundgedanke von SaaS Modellen ist, dass die erforderlichen Systemkapazitäten flexibel der Bedarfssituation folgen können. Hierdurch soll die Flexibilität geschaffen werden, Kapazitäten im Bedarfsfall kurzfristig zu erhöhen und gegebenenfalls später wieder zu reduzieren. Die Bereitstellung der entsprechenden Ressourcen obliegt dabei dem SaaS-Anbieter, weshalb diesbezügliche Aufwände und sprungfixe Kosten nicht durch die anwendenden Gesundheitsämter vorzuhalten sind.

- **Schnelle Implementierung:** Da es sich im Falle von SaaS um eine bereits entwickelte Software handelt, sind die Implementierungsaufwände im Vergleich zu Eigenentwicklungen oder lokal gehosteter Software vergleichsweise gering. SaaS-Modelle können je nach Bedarfsanpassung innerhalb weniger Wochen eingeführt werden.
- **IT-Sicherheit, Datensicherheit und Datenhoheit: Anbieter** mit zertifizierten Sicherheitsstandards (z. B. ISO 27001) können einen hohen Schutz bieten. Der nachhaltige Grad der Sicherheit ist allerdings von der dauerhaften Pflege und Wartung der Systeme und Software abhängig. Diese ist durch den SaaS Anbieter sicherzustellen. Da die Bereitstellung der SaaS Lösungen in der Regel innerhalb des Behördenumfelds stattfindet, ist eine Verlagerung auf externe Infrastruktur professioneller SaaS Anbieter (Hyperscaler) nicht zwingend erforderlich. In diesem Fall bewegen sich die Daten ausschließlich im Behördenumfeld, wodurch die Datenhoheit und -kontrolle sichergestellt werden kann.
- **Entlastung der IT:** Durch die vorgenannte Verlagerung von Aufgaben der IT der Gesundheitsämter zum SaaS Anbieter (insbesondere die Infrastruktur und Software-Entwicklung und Pflege betreffend), können sich die Gesundheitsämter stärker auf ihre Kernaufgaben konzentrieren. Die ansonsten benötigten Ressourcen können anderweitig genutzt werden.

Ist die Nutzung eines SaaS Modells im spezifischen Fall sinnvoll und besteht ein diesbezügliches Angebot im Behördenumfeld, sollten sich die Beteiligten hinsichtlich der Detailanforderungen (beispielsweise Anpassung des Datenmodells, technischer Schnittstellen etc.) entsprechend abstimmen. Diesbezügliche Aufwände können gegebenenfalls wiederum projekthaften Charakter haben.

Zu regeln sind jedoch vor allem die Leistungen, welche im Rahmen von SaaS erbracht werden sollen. Dabei ist insbesondere Wert auf Verlässlichkeit, Performance, Skalierbarkeit, Stabilität und Sicherheit zu legen. Diesbezügliche Vereinbarungen sind spezifisch zu regeln.

2.3.3 Fazit zur Nachnutzung von Open-Source Lösungen im Rahmen SaaS

Die Nachnutzung von OSS-Lösungen als SaaS im Umfeld der Gesundheitsämter kann sinnvoll sein. Dies ist eine Frage der individuellen Abwägung unter Berücksichtigung der zuvor beleuchteten Faktoren.

Eine wesentliche Herausforderung wird jedoch das Finden entsprechender Angebote in der vielfältigen Behördenlandschaft sein. Dies gilt vor allem vor dem Hintergrund, dass es sich potenziell nicht nur um ein spezifisches Verfahren handelt, sondern viele verschiedene Verfahren von den unterschiedlichste Ämtern angeboten werden können. Daher empfiehlt sich die Gründung einer übergreifenden Community, innerhalb derer abgestimmt wird, wer welche Entwicklungen durchführt und Dienste anbietet. Als Ergänzung zur Community kann auch die Implementierung eines Marktplatzes sinnvoll sein, insbesondere deshalb, um dort direkt angebotene Leistungen für Interessenten transparent darzustellen.

Hinweis: OSS ≠ automatisch souverän – SaaS ≠ automatisch effizient

Eine Open-Source-Lösung im SaaS-Modell, die von einem externen Anbieter betrieben wird, kann im schlechtesten Fall dieselben Abhängigkeiten erzeugen wie proprietäre Software. Umgekehrt kann eine proprietäre Lösung im Eigenbetrieb oder in einem öffentlich kontrollierten Rechenzentrum größere Souveränitätsgewinne ermöglichen. Entscheidend ist das Gesamtmodell – nicht allein die Lizenz.

2.4 Transformation bestehender Systeme zu Open Source

Viele im ÖGD eingesetzte Fachverfahren sind historisch gewachsen und basieren auf proprietärer Software, oft stark angepasst an die jeweiligen Anforderungen der Ämter. Die vollständige Ablösung durch eine Open-Source-Neuentwicklung ist in der Praxis häufig nicht realistisch – weder personell noch finanziell.

Ein möglicher Transformationspfad besteht in der schrittweisen Öffnung bestehender Systeme: Gemeinsam mit den bestehenden Anbietern kann geprüft werden, welche Komponenten eines Fachverfahrens perspektivisch offen lizenziert und zur gemeinschaftlichen Weiterentwicklung freigegeben werden können – etwa Schnittstellen, Datenmodelle oder Analysemodule.

Erfolgreiche Transformationsstrategien setzen dabei auf:

- Kooperation statt Konfrontation mit etablierten Anbietern
- Modulare Entflechtung bestehender Lösungen
- Vergabe neuer Entwicklungsanteile unter Open-Source-Lizenzen
- Pilotprojekte mit Rückkanal zur Community

Besonders sinnvoll ist ein schrittweiser Umbau bei Systemen, bei denen ohnehin Modernisierungsbedarf besteht – z. B. durch veraltete Technologien, fehlende Schnittstellen oder mangelhaften Support. In solchen Fällen kann eine Migration in OSS-Komponenten nicht nur technische, sondern auch strategische Vorteile bringen.

Langfristiges Ziel sollte nicht zwangsläufig die vollständige Ablösung proprietärer Systeme sein, sondern die Stärkung interoperabler, offener Architekturprinzipien, die digitale Souveränität ermöglichen – auch im Zusammenspiel mit bewährten Elementen aus bestehenden Lösungen.

3 SCHLUSSWORT

Dieses Positionspapier bietet eine fundierte Grundlage, um den Einsatz von OSS im ÖGD gezielt voranzutreiben. Es zeigt Vorteile auf und gibt praxisnahe Empfehlungen zur strategischen Nutzung. Das Fazit: Digitale Souveränität kann durch OSS nachhaltig gestärkt werden, erfordert aber gezielte Planung und Zusammenarbeit.

Ein besonderer Dank gilt den Mitarbeitenden der Gesundheitsämter, die mit ihrer Expertise und praktischen Erfahrung entscheidend zur Entwicklung dieses Papiers beigetragen haben. Ebenso danken wir der Arbeitsgruppe und den IT-Fachleuten für ihre wertvolle Unterstützung sowie den Interviewpartnern für ihre Einblicke in Herausforderungen und Chancen von OSS im ÖGD.

Die öffentliche Verwaltung in Deutschland investiert jährlich Millionen in OSS. Die Bundesregierung fördert OSS-Projekte über ZenDiS, während Bundesländer wie Schleswig-Holstein und Bayern eigene Strategien zur Reduzierung von Abhängigkeiten proprietärer Anbieter verfolgen.

Dies unterstreicht das wachsende Bewusstsein für OSS als nachhaltige und kosteneffiziente Lösung im öffentlichen Sektor.

Die hier formulierten Empfehlungen sollen Diskussionen und Entwicklungen im Bereich OSS im ÖGD anstoßen. Ziel ist es, nachhaltige Initiativen und Kooperationen zu fördern, um die Digitalisierung des öffentlichen Gesundheitsdienstes effizient und sicher voranzubringen.

Vielen Dank an alle Mitwirkenden!

4 ANHANG

4.1 Umfrage

1. Interesse an der Mitarbeit in den Arbeitsgruppen

Frage 1 Haben Sie Interesse, aktiv an der **Arbeitsgruppe Low Code** mitzuwirken?

- ☐ Ja, ich bin interessiert
- ☐ Vielleicht, ich benötige weitere Informationen
- ☐ Nein, ich bin nicht interessiert

Frage 2 Haben Sie Interesse, aktiv an der **Arbeitsgruppe Open Source (OSS)** mitzuwirken?

- ☐ Ja, ich bin interessiert
- ☐ Vielleicht, ich benötige weitere Informationen
- ☐ Nein, ich bin nicht interessiert

2. Motivationen und Erwartungen

Frage 3 Was motiviert Sie, an der **Arbeitsgruppe Low Code** teilzunehmen? (Mehrfachauswahl möglich)

- ☐ Interesse an der Weiterentwicklung von Low-Code-Lösungen im öffentlichen Sektor
- ☐ Möglichkeit, eigene Erfahrungen und Ideen einzubringen
- ☐ Netzwerkbildung mit anderen Experten im Gesundheitswesen und der öffentlichen Verwaltung
- ☐ Potenzial, innovative Lösungen für meine eigene Behörde zu finden
- ☐ Andere (bitte spezifizieren)

Frage 4 Was motiviert Sie, an der **Arbeitsgruppe Open Source (OSS)** teilzunehmen? (Mehrfachauswahl möglich)

- ☐ Interesse an der Förderung von Open-Source-Software in der Verwaltung
- ☐ Möglichkeit, eigene Erfahrungen und Ideen einzubringen
- ☐ Austausch über Best Practices im Umgang mit Open-Source-Lösungen
- ☐ Potenzial, Open-Source-Lösungen in meiner Behörde zu implementieren
- ☐ Andere (bitte spezifizieren)

3. Erfahrung und Expertise

Frage 5 Welche Erfahrungen oder Expertisen bringen Sie mit, die für die Arbeit in der **Low-Code-Arbeitsgruppe** von Nutzen sein könnten?

- ☐ Technische Expertise im Bereich Low-Code
- ☐ Erfahrung mit der Implementierung von Low-Code-Plattformen
- ☐ Kenntnis der Anforderungen und Prozesse im öffentlichen Sektor
- ☐ Andere (bitte spezifizieren)

Frage 6 Welche Erfahrungen oder Expertisen bringen Sie mit, die für die Arbeit in der **Open-Source-Arbeitsgruppe** von Nutzen sein könnten?

- ☐ Technische Expertise im Bereich Open Source
- ☐ Erfahrung mit der Implementierung von Open-Source-Lösungen
- ☐ Kenntnis der rechtlichen und Lizenzfragen von Open Source
- ☐ Andere (bitte spezifizieren)

4. Bedenken und Herausforderungen

Frage 7 Welche potenziellen Herausforderungen sehen Sie bei der Teilnahme an der **Arbeitsgruppe Low Code**? (Mehrfachauswahl möglich)

- ☐ Zeitliche Einschränkungen
- ☐ Geringes internes Interesse an Low-Code-Themen in meiner Behörde
- ☐ Fehlende technische Expertise
- ☐ Andere (bitte spezifizieren)

Frage 8 Welche potenziellen Herausforderungen sehen Sie bei der Teilnahme an der **Arbeitsgruppe Open Source (OSS)**? (Mehrfachauswahl möglich)

- ☐ Zeitliche Einschränkungen
- ☐ Geringes internes Interesse an Open-Source-Themen in meiner Behörde
- ☐ Fehlende technische Expertise
- ☐ Andere (bitte spezifizieren)

5. Thematische Schwerpunkte pro Arbeitsgruppe

Frage 9 Zu welchen spezifischen Themenfeldern möchten Sie in der **Arbeitsgruppe Low Code** mitwirken? (Mehrfachauswahl möglich)

- ☐ Einsatzmöglichkeiten von Low-Code-Plattformen in der Verwaltung
- ☐ Integration von Low-Code-Lösungen in bestehende IT-Systeme
- ☐ Sicherheit und Compliance bei Low-Code
- ☐ Schulungen und Skill-Building für Low-Code-Plattformen
- ☐ Zusammenarbeit mit externen Anbietern und Dienstleistern

Frage 10 Zu welchen spezifischen Themenfeldern möchten Sie in der **Arbeitsgruppe Open Source (OSS)** mitwirken? (Mehrfachauswahl möglich)

- ☐ Rechts- und Lizenzfragen bei Open Source
- ☐ Nachhaltigkeit und Wartung von Open-Source-Projekten
- ☐ Beschaffung und Implementierung von Open-Source-Software
- ☐ Sicherheitsaspekte von Open-Source-Software
- ☐ Kooperation und Wissensaustausch zwischen Behörden bei Open Source

Frage 11 Was möchten Sie uns noch sagen?

4.2 Quellen

Öffentliche IT. Wird die Vergabereform die Tür für mehr Open Source in der Verwaltung öffnen?

Verfügbar unter: https://www.oeffentliche-it.de/blog/wird-die-vergabereform-die-tuer-fuer-mehr-open-source-in-der-verwaltung-oeffnen/

Behörden Spiegel. Open Source Software im Vergaberecht
Verfügbar unter: https://www.behoerden-spiegel.de/2024/06/05/open-source-software-im-vergaberecht/

openCoDE. Open-Source-Plattform der öffentlichen Verwaltung
Verfügbar unter: https://opencode.de/

OSB Alliance. Strategie für die Zukunft: Open Source in der Verwaltung
Verfügbar unter: https://osb-alliance.de/featured/strategie-fuer-die-zukunft-open-source-in-der-verwaltung

CIO Bund. Open Source und digitale Souveränität
Verfügbar unter: https://www.cio.bund.de/Webs/CIO/DE/digitale-loesungen/digitale-souveraenitaet/open-code/open-code-node.html

Open Government Deutschland. Open-Source-Plattform der öffentlichen Verwaltung
Verfügbar unter: https://www.open-government-deutschland.de/opengov-de/ogp/aktionsplaene-und-berichte/3-nap/eine-open-source-plattform-der-oeffentlichen-verwaltung-1981538

Zentrum für Digitale Souveränität (ZenDiS)
Verfügbar unter: https://zendis.de/

4.3 Glossar

Begriff	Definition
Bier-Lizenz	Eine humorvolle Open-Source-Lizenz, die vorschreibt, dass Nutzer dem Entwickler ein Bier ausgeben müssen, wenn sie ihn treffen.
Cloud Computing	Bereitstellung von IT-Ressourcen wie Speicherplatz, Rechenleistung oder Anwendungen über das Internet.
Community	Eine Gruppe von Entwicklern und Nutzern, die gemeinsam an Open-Source-Projekten arbeiten, Fehler beheben und Weiterentwicklungen vorantreiben.
Copyleft	Eine Lizenzbedingung, die vorschreibt, dass abgeleitete Werke ebenfalls unter der gleichen Open-Source-Lizenz veröffentlicht werden müssen. Lizenzen mit Copyleft-Effekt sind z. B. GPL, AGPL oder LGPL.
Cyber-Resilienz	Die Fähigkeit von IT-Systemen, Angriffe abzuwehren und den Betrieb trotz Cyber-bedrohungen aufrechtzuerhalten.
Digitale Souveränität	Fähigkeit von Individuen, Organisationen und öffentlichen Institutionen, ihre Rolle in der digitalen Welt selbstständig, selbstbestimmt und sicher auszuüben. Sie umfasst unter anderem: • Datensouveränität: Kontrolle über Speicherung und Verarbeitung sensibler Daten • Technologische Unabhängigkeit: Wahlfreiheit bei Infrastruktur, Software, Anbietern • Transparenz & Kontrolle: Nachvollziehbarkeit digitaler Abläufe
Fork	Eine Abspaltung eines Open-Source-Projekts, bei der eine alternative Version weiter-entwickelt wird.
GNU/GPL	Eine der bekanntesten Open-Source-Lizenzen, die sicherstellt, dass Software frei nutzbar, veränderbar und weiterverbreitbar bleibt.
Informationssicherheit	Schutz vor unberechtigtem Zugriff, Ausfällen, Manipulation (Definition in Anlehnung an den Beauftragten der Bundesregierung für Informationstechnik)
Interoperabilität	Die Fähigkeit verschiedener Systeme oder Softwarelösungen, miteinander zu kommunizieren und Daten auszutauschen.
Low-Code-Entwicklung	Eine Softwareentwicklungsmethode, die wenig manuelle Programmierung erfordert und auf visuelle Modellierung setzt.
NIS2-Richtlinie	Eine EU-weite Regelung zur Stärkung der Cybersicherheit in kritischen Infrastrukturen, auch relevant für Open Source.
Open Government	Ein Ansatz zur Förderung von Transparenz, Partizipation und Zusammenarbeit durch den Einsatz offener Technologien und Daten.
Open Source Program Office (OSPO)	Eine interne Einheit in Organisationen zur Verwaltung und strategischen Nutzung von Open-Source-Software.
Open Source Software (OSS)	Software, deren Quellcode offen zugänglich ist, modifiziert und weiterverbreitet werden darf. OSS lässt sich grob in zwei Kategorien unterteilen: Permissive Lizenzen und Copyleft-Lizenzen.
Permissive Lizenz	Permissive Lizenzen erlauben eine weitgehend freie Nutzung, auch in proprietärer Software, oft nur mit der Anforderung, den ursprünglichen Lizenztext beizulegen. Permissive Lizenzen sind z. B. MIT, Apache 2.0 oder BSD.
Public Code	Software, die speziell für den öffentlichen Sektor entwickelt und unter einer Open-Source-Lizenz freigegeben wird.

Begriff	Definition
SBOM (Software Bill of Materials)	Eine detaillierte Liste aller Komponenten einer Software, inklusive Abhängigkeiten und Lizenzinformationen.
SaaS (Software-as-a-Service)	Ein Bereitstellungsmodell, bei dem Software über das Internet genutzt wird, ohne lokal installiert zu werden.
Total Cost of Ownership (TCO)	Die Gesamtkosten einer Softwarelösung über ihren gesamten Lebenszyklus, einschließlich Entwicklung, Wartung und Support.
Transparenz	Offenlegung von Quellcode, Prozessen und Kosten im Zusammenhang mit Open-Source-Software, um Nachvollziehbarkeit und Kontrolle zu gewährleisten.
Vergaberecht	Gesetzliche Regelungen zur Beschaffung von IT-Lösungen im öffentlichen Sektor, die auch für Open-Source-Software relevant sind.

Auftraggeberin

Vorsitzland UAG Digitalisierung der AG ÖGD der AOLG
Senatsverwaltung für Wissenschaft, Gesundheit und Pflege
Oranienstraße 106
10969 Berlin

E-Mail: gst-uagdigitalisierung@paktoegd.de

Ansprechpartnerin UAG

Frau Dr. Sabine Hermann
Vorsitzende UAG Digitalisierung der AG ÖGD der AOLG

Erstellt durch die Geschäftsstelle der UAG Digitalisierung

Dienstleister (im Auftrag)

Alexander Barion
Philipp Menze

HISOLUTIONS

HiSolutions AG
Schloßstraße 1
12163 Berlin

E-Mail: info@hisolutions.com

Stand: 25.04.2025