

Untersuchung von Open-Source-Mängelmeldern für den Einsatz in einer Kommunalverwaltung

Evaluation of Open Source Public Service Concerns Reporting Systems

Lukas Haas¹, Florian Knoch¹

¹Fachbereich Technik, Hochschule Mainz · lukas.haas@students.hs-mainz.de

Zusammenfassung: In diesem Beitrag wird die Eignung verschiedener auf dem Markt erhältlicher Open-Source-Mängelmelderlösungen für den Einsatz in einer Kommunalverwaltung untersucht. Auf der Grundlage einer Literaturrecherche wird ein Kriterienkatalog entwickelt, mithilfe dessen die Plattformen qualitativ und quantitativ evaluiert und miteinander verglichen werden. Vor der Evaluation werden die Plattformen individuell auf einem Server eingerichtet. Zur Bewertung der Kriterien wird für jede Plattform sowohl diese eigens installierte als auch eine bereits umgesetzte, im Internet verfügbare Instanz genutzt. Die Plattformen werden auf der Grundlage des Kriterienkataloges bewertet und die Vor- und Nachteile der einzelnen Lösungen herausgearbeitet. In einer abschließenden Beurteilung wird eine Nutzungsempfehlung für eine der Plattformen abgegeben.

Schlüsselwörter: Mängelmelder, Softwarevergleich, Kommunalverwaltung

Abstract: *This paper examines the suitability of various Open Source Public Service Concerns Reporting Systems available on the market for use in a local government administration. Based on literature research, a catalogue of criteria is developed, with the help of which the platforms are qualitatively and quantitatively evaluated and compared with each other. Before the evaluation, the platforms are set up individually on a server. For the evaluation of the criteria, both this specially installed instance and an already implemented instance available on the Internet will be used. The platforms are evaluated based on the criteria catalogue and the advantages and disadvantages of the individual solutions are worked out. In a final assessment, a recommendation for use is made for one of the platforms.*

Keywords: *Public Service Concerns Reporting System, software comparison, local government*

1 Einleitung

Im Rahmen der Digitalisierung deutscher Stadtverwaltungen sind in den vergangenen Jahren zunehmend neue Konzepte für verschiedene Themenbereiche des Verwaltungshandelns entwickelt und von vielen Städten bereits umgesetzt worden. Ein leicht realisierbares und wirkungsvolles Konzept ist die Idee eines Mängelmelders. Vaerst (2017, p. 101) sieht Mängelmelder als vertrauensbildende Maßnahme und führt aus, dass der Mängelmelder „ein erster Schritt zu einer neuen Kultur der intensiveren Bürgerbeteiligung [sein] kann“. Auf einer zentralen Plattform können Bürgerinnen und Bürger, denen Mängel an öffentlichen Orten auffallen, diese beschreiben, lokalisieren und so der Kommunalverwaltung mitteilen. Auf derselben Plattform entsteht dann eine Historie für jeden Mangel.

Auf dem Markt existieren verschiedene Konzepte zur Umsetzung von Mängelmelderlösungen, die für Kommunen und Bürger nutzbar sind. Während manche Anbieter landesweite Plattformen anbieten, bei denen Mängel auf dem Gebiet mehrerer Verwaltungseinheiten oder

ganzer Staatsgebiete gemeldet werden können, entwickeln andere Anbieter Lösungen, die auf das Gebiet einer Kommune begrenzt sind und von dieser selbst betreut werden. Bei dem ersten Konzept wird die Plattform zentral beim Anbieter gehostet, interessierte Kommunen können sich dort registrieren. Bei dem zweiten Konzept wird der Mängelmelder allein von der Kommunalverwaltung verwaltet und selbst gehostet.

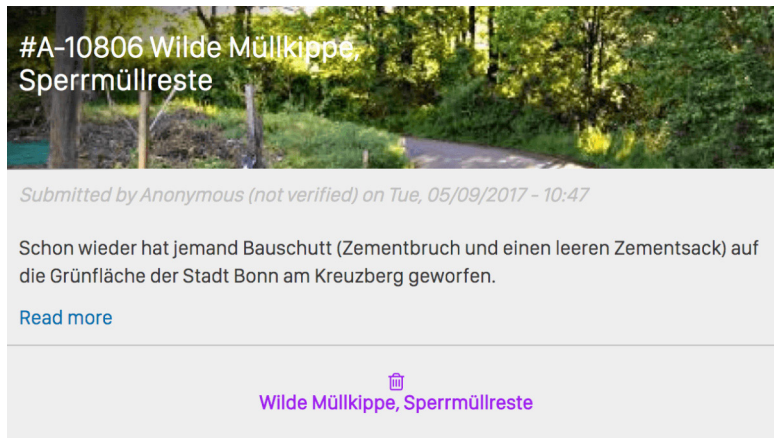


Abb. 1: Mangelmeldung mit Fotografie und Beschreibung in der Plattform Mark-a-Spot (Kreis, H. (2022): *Mark-a-Spot Webauftritt*. Retrieved Jan 30, 2022, from <https://markaspot.de/images/screen-requests.png>)

Für ein solches kommunales Anliegenmanagement existieren verschiedene Plattformen, die sich in Kosten und Umfang der Leistung und Betreuung teilweise stark unterscheiden. In der vorliegenden Arbeit sollen nur Plattformen betrachtet werden, die sowohl von einer Stadt, Gemeinde oder Kommune selbst gehostet werden können als auch kostenfrei und quelloffen sind. Auf dieser Grundlage werden folgende Plattformen für die Untersuchung ausgewählt.

Die Plattform „FixMyStreet“ ist eine Anliegenmanagementplattform der britischen Firma MySociety. „FixMyStreet“ existiert seit 2007 und bietet eine Lösung an, die lokal von Kommunalverwaltungen installiert werden kann. Die Kommunalverwaltung kann selbst entscheiden, ob sie die Plattform selbst betreut, oder einen Servicevertrag mit MySociety abschließt.

Die seit 2011 entwickelte Plattform „Mark-a-Spot“ bietet ebenfalls eine Anliegenmanagementlösung. Die Plattform ist quelloffen und basiert auf dem Contentmanagementsystem Drupal.

Bei der dritten untersuchten Lösung handelt es sich um die Plattform „Klarschiff“. Die durch die Hanse- und Universitätsstadt Rostock initiierte und durch das Fraunhofer-Institut für Graphische Datenverarbeitung in Rostock seit 2012 entwickelte, quelloffene Lösung kann ebenfalls von Kommunen kostenfrei installiert werden.

2 Erstellung des Kriterienkatalogs

Die Evaluation von Software folgt nach Hegner (2003, p. 8) einem vierstufigen Schema, an dessen Beginn die Formulierung von Wertekriterien steht. Hegner definiert diese als „Kriterien, die das Produkt erfüllen muss, um als gut, wertvoll [...] gelten zu können“. In der zweiten Phase folge die Operationalisierung der Kriterien. Mithilfe der Operationalisierung werde ein Leistungsstandard festgelegt, den ein Produkt erfüllen muss, damit das Kriterium erfüllt ist. Die anschließende Analyse beinhaltet dann den Vergleich der verschiedenen Lösungen durch Messung und Vergleich mit den vorgegebenen Leistungsstandards. In der abschließenden Phase „werden die verschiedenen Ergebnisse zu einem einheitlichen Werturteil verknüpft“ (Hegner, 2003, p. 9).

Anforderungen an eine Softwarelösung können in verschiedene Kategorien unterteilt werden. Bracket (1990) unterscheidet hierbei unter anderem funktionale und nicht-funktionale Anforderungen. Eine funktionale Anforderung ist in diesem Zusammenhang eine Funktionalität, die das System erfüllen soll. Eine nicht-funktionale Anforderung hingegen ist eine Eigenschaft des Systems, die nicht direkt mit der Funktionalität zusammenhängt.

2.1 Funktionale Anforderungen

Dias et al. (2017) definierten auf der Grundlage einer Umfrage in Sri Lanka Anforderungen, die ein Mängelmeldersystem erfüllen soll. In Berntzen et al. (2018) werden unter anderem die Gemeinsamkeiten von Mängelmeldersystemen beleuchtet und es wird die Entwicklung einer Mängelmelderlösung für die Stadt Wiesbaden beschrieben. Daraus können ebenfalls Anforderungen abgeleitet werden, die ein solches System erfüllen soll. Vaerst (2017) erläutert das Konzept Mängelmelder und stellt verschiedene Lösungen und deren Funktionalitäten vor. Choudhury (2012) untersucht die lokale Mängelmelderlösung „FixEmUp“ der Stadt Tornio und definiert ebenfalls Anforderungen, die die Plattform erfüllen soll.

Anhand dieser Quellen werden zunächst Hauptanforderungen aufgestellt. Sie beschreiben die Grundfunktionalitäten des Systems und werden in mehreren Quellen genannt. Neben den Hauptanforderungen werden Nebenanforderungen definiert. Die Nebenanforderungen setzen sich sowohl aus Kriterien zusammen, die in mindestens einer der Quellen genannt werden als auch aus solchen, die aus den eigenen Überlegungen der Autoren stammen.

2.1.1 Hauptanforderungen

- *Geo-Tag-Feature*: Die Mängelmelderlösung soll eine Funktionalität enthalten, mit der die eigene Position des (mobilen) Endgerätes abgefragt werden kann (Dias et al., 2017; Berntzen et al., 2018; Vaerst, 2017).
- *Fotografie*: Bei der Erhebung eines Mangels soll es möglich sein, eine Fotografie der Mangelstelle anzufügen (Dias et al., 2017; Berntzen et al., 2018; Vaerst, 2017; Choudhury, 2012).
- *Beschreibung*: Bei der Erhebung des Mangels soll dem Nutzer die Möglichkeit eingeräumt werden, den Mangel durch freitextuelle Beschreibung spezifizieren zu können (Dias et al., 2017; Berntzen et al., 2018; Choudhury, 2012).
- *Kategorisierung*: Die Mängel sollen in verschiedene Kategorien eingeteilt werden können. Diese sollen sinnvollerweise auf die Arbeitsbereiche verschiedener kommunaler

Stellen anpassbar und frei definierbar sein (Dias et al., 2017; Berntzen et al., 2018; Vaerst, 2017).

- *Dokumentation des Mangels*: Der Mangel soll im System lückenlos und transparent dokumentiert werden. Der Bürger soll bestehende Meldungen ansehen können. Die Dokumentation des Bearbeitungsstandes kann beispielsweise durch ein Ampelsystem realisiert werden, bei dem die verschiedenen Farben für verschiedene Bearbeitungsphasen stehen (Dias et al., 2017; Vaerst, 2017).
- *Information des Melders*: Der Bürger, der den Mangel gemeldet hat, soll über die weitere Bearbeitung des Mangels informiert werden. Dies kann durch E-Mail-Benachrichtigung erfolgen, wenn sich der Bearbeitungsstand ändert (Dias et al., 2017; Berntzen et al., 2018; Vaerst, 2017).
- *Manuelle Erfassung*: Der Nutzer oder ein Sachbearbeiter der Stadt soll in der Lage sein, einen Mangel nicht vor Ort, sondern am Computer zu melden. Es soll eine Möglichkeit zur Wahl der Position des Mangels auf einer Kartenanwendung gegeben sein. Dies kann durch einen Gazetteer-Dienst unterstützt werden (Vaerst, 2017; Choudhury, 2012).
- *Kartendarstellung*: Die Mängel sollen übersichtlich mithilfe von Markern auf einer interaktiven Karte verzeichnet sein. Zwischen verschiedenen Kartengrundlagen kann gewählt werden. Eigene Web-Map-Services können genutzt werden (Vaerst, 2017; Choudhury, 2012).

2.1.2 Nebenanforderungen

- *Priorisierung*: Die Mängel sollen priorisiert werden können. Dies kann manuell oder automatisiert anhand der Kategorisierung erfolgen. Die Priorisierung kann auf der Website des Mängelmelders veröffentlicht oder rein intern vermerkt werden (Dias et al., 2017).
- *Dopplungsfilter*: Mithilfe eines Dopplungsfilters können Mehrfachmeldungen für denselben Mangel vermieden werden. Der Dopplungsfilter soll automatisch Meldungen derselben Kategorie, die innerhalb eines bestimmten Umkreises liegen und innerhalb einer bestimmten Zeitspanne gemeldet wurde, als Duplikate erkennen (Vaerst, 2017).
- *Administrationsbereich*: Die Plattform soll über einen Administrationsbereich verfügen, in dem alle relevanten Bearbeitungsschritte vorgenommen werden können. Darunter zählt die Verwaltung der Mängel, das Anlegen verschiedener Nutzerrollen und Institutionen, die Aktualisierung der Status und das Anpassen der Seiteninhalte (Vaerst, 2017).
- *Offene Standards*: Der Mängelmelder soll Standards zum Datentransfer unterstützen. Darunter zählt der offene Standard für das Anliegenmanagement zwischen Behörden und Bürgern „Open311“ und das zugehörige Austauschformat „GeoReport v2“. Ein in Deutschland etablierter Standard ist XÖV. Er dient dem Datenaustausch zwischen öffentlichen Behörden und kann als Datentransferformat für Mängelmelder dienen (Vaerst, 2017).
- *Datenbank*: Die Mängel sollen in einer systeminternen Datenbank geführt werden. In dieser Datenbank können auch Institutionen und Nutzer angelegt werden. Die Historie der Datenbank kann für Auswertungen genutzt werden.
- *Navigationsfunktion*: Mithilfe der Navigationsfunktion wird eine Route von der aktuellen Position des Endgerätes zum Mangel berechnet und in der Karte angezeigt.
- *Dashboard / Statistiken*: Die Kommunalverwaltung soll in der Lage sein, die Mangelhistorie nach verschiedenen Kriterien statistisch auswerten zu können. Die Plattform soll die Möglichkeit bieten, die Statistiken automatisiert zu Diagrammen oder zu einem Dashboard aufzubereiten und dem Nutzer verfügbar zu machen.

2.2 Nicht-funktionale Anforderungen

Neben den Funktionalitäten, die die Plattform erfüllen muss, existieren weitere Anforderungen, die an die Plattform oder deren Dokumentation gestellt werden. Diese nicht-funktionalen Anforderungen behandeln zumeist weniger die spezifischen Forderungen der Anwendungsdomäne, sondern bewerten die Art und Weise, auf die die Lösung umgesetzt wurde. Dias et al. (2017) haben eine Reihe an nicht-funktionalen Anforderungen an eine Mängelmelderplattform aufgestellt. Um die recht grob umrissenen Begriffe genauer zu definieren, wurden weitere Quellen zur Spezifizierung der Anforderungen gesichtet. Jackson et al. (2011) stellen auf Grundlage der ISO 25010 einen Katalog auf, mit dessen Hilfe Softwarepakete kriterienbasiert evaluiert werden können. Abstrakte Begriffe werden in einzelne Fragestellungen aufgeteilt und operationalisiert, sodass sie direkt an der Plattform getestet werden können. Folgende Anforderungen werden verglichen:

- *Usability*: Unter der Usability wird die Nutzbarkeit einer Lösung verstanden. Die Fragestellungen müssen vor dem Hintergrund evaluiert werden, welche Zielgruppe mit einem Mängelmelder angesprochen wird, und welche Grundkenntnisse vorausgesetzt werden können. Da bei einem Mängelmelder die gesamte Bürgerschaft angesprochen wird, müssen die Basisfunktionalitäten leicht erkennbar und der Ablauf der Mängelmeldung verständlich und intuitiv sein (ISO 25010; Jackson et al., 2011).
- *Dokumentation und Buildability*: Da der Mängelmelder von der Kommunalverwaltung selbst gehostet werden soll, ist eine ausführliche Installations- und Nutzungsanleitung wichtig für die Nutzung und die Anpassung der Einstellungen auf Themen, die für die lokale Kommune wichtig sind. Hierbei wird der Aufbau und Umfang der Dokumentation bewertet. Zudem fließen Erfahrungen der eigenen Installation der Plattform ein (Jackson et al., 2011).
- *Flexibility*: Unter der Flexibilität der Plattform wird verstanden, wie leicht die Standardeinstellung der Lösung auf lokale Belange angepasst werden kann. Flexibilität bedeutet für diese Untersuchung, wie leicht neue Mängelkategorien und neue Formularfelder bei der Mängelerfassung angefügt werden können (ISO 25010; Jackson et al., 2011).
- *Security*: Unter der Security der Lösung versteht man den Datenschutz in der Datenbank. Bewertet wird, ob die Daten bereits durch Installation der Plattform gesichert sind, oder ob externe Schutzmaßnahmen notwendig sind (ISO 25010).

3 Evaluation der Plattformen

Mithilfe des erstellten Kriterienkataloges werden die drei Plattformen evaluiert. Es wird überprüft, inwiefern die Anforderungen erfüllt sind. Hierfür wird jedes Kriterium in vier Abstufungen quantisiert (– – ; – ; + ; ++). Kann ein Kriterium nicht überprüft werden, erhält es die Wertung „n“ und nimmt nicht an der Gesamtauswertung teil. Die Struktur der Ergebnisse folgt der Strukturierung der Anforderungen.

Die Grundfunktionalitäten der Plattformen werden bewertet, indem die einzelnen Kriterien in einer selbst installierten Instanz getestet werden. Zusätzlich werden die Möglichkeiten einer voll funktionsfähigen Umsetzung anhand der Nutzersegmente bestehender Instanzen analysiert.

3.1 Installation

Die Plattformen werden zu Testzwecken installiert, um die Administration überprüfen und um den Installationsprozess bewerten zu können. Vor dem Hintergrund, dass nicht jede Kommune einen erfahrenen Systemadministrator beschäftigen kann, wird die Installation von den Autoren durchgeführt, die lediglich über Grundkenntnisse im Umgang mit Linux-Servern verfügen. Als Zeitlimit werden pro Lösung ein Arbeitstag (acht Stunden) veranschlagt, sollte die Installation in dieser Zeit nicht gelingen, gilt der Installationsversuch als gescheitert. Für die Installation wird eine auf Linux/Ubuntu basierende virtuelle Maschine einer Server-Instanz eingesetzt. Die Installation der Systeme erfolgt nach Einrichtung des jeweiligen Projektverzeichnisses manuell über die Konsole mithilfe der in den Installationsdokumentationen vorgegebenen Befehlen. Bei allen Paketen wird es im Laufe des Installationsprozesses nötig, Drittsoftware zu installieren. Bei „Mark-a-Spot“ handelt es sich hier um ein Apache-Modul, ein php-Modul, „MySQL“ und die Software „composer“. Bei „FixMyStreet“ wird zusätzlich „PostgreSQL“ benötigt. Für die Installation von Klarschiff werden im Laufe des Installationsprozesses „Node.js“, „npm“ und „Ruby“ benötigt. Während die Installation von „FixMyStreet“ ohne Probleme durchführbar war, traten bei „Mark-a-Spot“ Versionskompatibilitätsprobleme bezüglich der benötigten Drittsoftware auf, die durch einige Anpassungen gelöst werden konnten. Bei der Installation von „Klarschiff“ schlug die Einrichtung des Proxys sowie die Installation von Zusatzpaketen fehl. Der Administrationsbereich dieser Plattform kann somit nicht bewertet werden, diese Funktionalitäten werden mit der Wertung „n“ versehen.

3.2 Funktionale Anforderungen

Die Tabelle 1 zeigt das Ergebnis der Überprüfung der Hauptanforderungen. Es wurde sowohl die Umsetzung des Kriteriums in der eigenen Instanz als auch die Möglichkeit der Umsetzung des Kriteriums mithilfe des Nutzersegments einer bereits existierenden deutschsprachigen Instanz bewertet. Der beschreibende Text dieses Kapitels hebt besondere Eigenschaften der Plattformen hervor und gibt einen knappen Einblick in die Ergebnisse des Vergleichs.

Tabelle 1: Vergleich der Lösungen anhand der Hauptanforderungen

Anforderung	FixMyStreet	Mark-a-Spot	Klarschiff
Geo-Tag-Feature	++	++	++
Fotografie	++	++	++
Beschreibung	+	+	++
Kategorisierung	-	+	++
Dokumentation des Mangels	++	++	+
Information des Melders	++	+	+
Manuelle Erfassung	++	++	++
Kartendarstellung	++	+	+

Die Evaluation der Hauptanforderungen ergab, dass die Erstellung neuer Mangelkategorien in der Plattform „FixMyStreet“ auf dem Server vorgenommen werden muss, während dies bei „Mark-a-Spot“ im Administrationsbereich vorgenommen werden kann. Die Kategorisie-

rung von Meldungen ist in den Plattformen teilweise unterschiedlich gelöst. Bei „Klarschiff“ existieren drei Kategorieebenen. In einem ersten Schritt kann gewählt werden, ob eine Idee oder ein Problem gemeldet werden soll, danach kann eine Oberkategorie und eine abhängige Unterkategorie gewählt werden. Die anderen Lösungen verfügen nur über eine Kategorieebene. Des Weiteren wurde festgestellt, dass zur Dokumentation des Mangels sowohl bei „FixMyStreet“ als auch bei „Mark-a-Spot“ im Administrationsbereich verschiedene Bearbeitungsstände gewählt werden können, die im Nutzersegment zu dem Mangel veröffentlicht werden. Im weiteren Verlauf der Evaluation stellte sich heraus, dass bei „FixMyStreet“ nicht nur der Melder des Mangels bei einer Änderung des Status per E-Mail informiert wird. Es können hier auch weitere Nutzer informiert werden, die sich für diesen Bereich (Straße, Wohnblock) registrieren. Bereits nach der Erstinstallation werden zudem bei „FixMyStreet“ individuelle Marker für die Lokalisierung der Mängel genutzt, die je nach Bearbeitungsstand unterschiedliche Farben annehmen.

Tabelle 2: Vergleich der Lösungen anhand der Nebenanforderungen

Anforderung	FixMyStreet	Mark-a-Spot	Klarschiff
Priorisierung	++	–	–
Dopplungsfilter	--	++	--
Administrationsbereich	+	++	n
Offene Standards	+	+	++
Datenbank	++	+	–
Navigationsfunktion	++	--	--
Statistiken/Dashboard	–	++	++

Bei der Evaluation der Nebenanforderungen zeigte sich, dass es lediglich in der Plattform „FixMyStreet“ vorgesehen ist, Mängel anhand der Kategorie zu priorisieren. Demgegenüber ist „Mark-a-Spot“ die einzige Lösung, bei der ein automatisierter Dopplungsfilter integriert ist. Daneben wurde festgestellt, dass sich die Administratorenbereiche der Plattformen hinsichtlich der Konzeption unterscheiden. Der Administrationsbereich der Plattform „FixMyStreet“ ist nur für die Bearbeitung der gemeldeten Mängel konzipiert. Die Anpassung der Benutzeroberfläche und alle Einstellungen (zum Beispiel das Hinzufügen neuer Kategorien) müssen auf dem Ubuntu-Server vorgenommen werden. Im Administrationsbereich der Plattform „Mark-a-Spot“ können, sofern die Nutzerberechtigung besteht, die gesamten Seiteninhalte und das Design angepasst werden. Die Überprüfung, ob und in welcher Weise offene Standards zum Datenaustausch auf den Plattformen genutzt werden, ergab, dass alle drei Systeme den offenen Standard Open311 unterstützen. „Klarschiff“ unterstützt zusätzlich das Austauschformat XÖV. Eine Besonderheit der Plattform „FixMyStreet“ ist die dort implementierte Navigationsfunktion, mithilfe derer sich der Nutzer zur Position des Mangels navigieren lassen kann. Im weiteren Verlauf der Evaluation wurden die Plattformen auch hinsichtlich der Möglichkeiten zur statistischen Auswertung und Aufbereitung der Daten überprüft. „Klarschiff“ und „Mark-a-Spot“ bieten weitreichende Statistikfunktionalitäten. Es können Auswertungen hinsichtlich Mangelarten, der Kategorien und der Bearbeitungsstatus erfolgen und im Nutzersegment in Form von Diagrammen zur Verfügung gestellt werden. Bei „Klarschiff“ werden mithilfe einer Heatmap die Standorte, an denen die meisten Mängelmel-

dungen innerhalb eines bestimmten Zeitraumes erfolgten, in einer interaktiven Kartenanwendung visualisiert.

3.3 Nicht-funktionale Anforderungen

In Tabelle 3 werden die nicht-funktionalen Anforderungen untersucht und gegenübergestellt.

Tabelle 3: Vergleich der Lösungen anhand der nicht-funktionalen Anforderungen

Anforderung	FixMyStreet	Mark-a-Spot	Klarschiff
Usability	++	++	+
Sinn der Plattform verständlich?	++	++	–
Basisfunktionalitäten erkennbar?	++	++	++
Prozessschritte verständlich?	+	++	++
Kernkomponente leicht erlernbar?	++	+	++
Nutzeranleitung verfügbar?	++	+	–
Grafische Gestaltung	++	+	++
Dokumentation und Buildability	+	+	–
Installationsanleitung verständlich?	++	+	--
Aufteilung in einzelne Abschnitte?	++	+	n
Für Nutzer verständlich?	++	+	--
Aufgabenorientiert? (Step-by-Step)	++	+	n
Sind Readmes vorhanden?	+	+	+
Klare Versionsdokumentation?	–	–	n
Benötigte Abhängigkeiten aufgeführt?	–	–	n
Benötigte Abhängigkeiten verfügbar?	–	–	n
Flexibility	+	++	n
Einfaches Anfügen neuer Kategorien	+	++	n
Security	–	–	n
Datenschutz in der Datenbank?	–	–	n

Um die nicht-funktionalen Anforderungen zu evaluieren, wurden auch die Dokumentationen der Plattformen auf den jeweiligen Websites betrachtet. „FixMyStreet“ und „Mark-a-Spot“ bieten eine sehr ausführliche Installationsanleitung, bei der jedoch einige notwendige zusätzliche Installationen nicht beschrieben werden. Die Installationsanleitung der Lösung „Klarschiff“ besteht lediglich aus einem Readme-Dokument im Git-Repositoryum. Die Plattform konnte mit dieser Anleitung nicht innerhalb der gegebenen Zeit installiert werden. Daneben wurde getestet, wie einfach es ist, individuelle Mangelkategorien hinzuzufügen. Diese strukturelle Anpassung von „Mark-a-Spot“ kann im Administrationsbereich erfolgen. Dies ist komfortabler als bei „FixMyStreet“, dort müssen diese Anpassungen auf der virtuellen Maschine vorgenommen werden.

4 Zusammenfassung und Nutzungsempfehlung

In diesem Beitrag werden Open-Source-Mängelmelder kriterienbasiert untersucht und miteinander verglichen. Auf Grundlage einer Literaturrecherche werden funktionale und nicht-funktionale Anforderungen an einen Mängelmelder definiert und operationalisiert. Die drei auf dem Markt angebotenen Open-Source-Plattformen „FixMyStreet“, „Mark-a-Spot“ und „Klarschiff“ werden auf einem Ubuntu-Server installiert und anhand des Kriterienkataloges vergleichend evaluiert.

Es wird gezeigt, dass „Mark-a-Spot“ und „FixMyStreet“ innerhalb eines Tages mit Ubuntu-Grundkenntnissen installiert werden können, dies ist bei „Klarschiff“ aktuell nicht der Fall. Kriterien, die sich auf den Administrationsbereich von „Klarschiff“ beziehen, können deshalb nicht überprüft werden. Die Lösungen „FixMyStreet“ und „Mark-a-Spot“ erzielen in der Kriterienauswertung bessere Ergebnisse als „Klarschiff“. „Mark-a-Spot“ und „FixMyStreet“ sind hinsichtlich der Konfiguration der Plattform nach der Installation verschieden konzipiert. Während bei „FixMyStreet“ alle Einstellungen auf dem Ubuntu-Server vorgenommen werden müssen, kann dies bei „Mark-a-Spot“ im Administrationsbereich der Website geschehen. Bei „Mark-a-Spot“ und „FixMyStreet“ ist es möglich, eigene Kartenlayer einzubinden, bei „Mark-a-Spot“ ist es leicht möglich neue Formularfelder in die Erfassungsmaske zu integrieren.

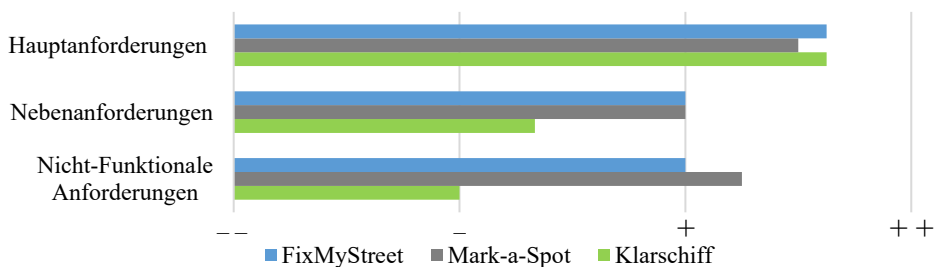


Abb. 2: Auswertung der Evaluation nach Anforderungen

Sowohl nach der Auswertung aller Kriterien als auch der Installation der Plattformen auf dem Server überzeugen die Lösungen „FixMyStreet“ und „Mark-a-Spot“. „Klarschiff“ liegt in der Gesamtwertung hinter den beiden anderen Lösungen. Nach Auswertung der Administrationsbereiche erweist sich „Mark-a-Spot“ als komfortabler und flexibler als „FixMyStreet“. Für den Einsatz einer Mängelmelderplattform in einer Kommunalverwaltung kann die vorliegende Untersuchung als Entscheidungsgrundlage dienen und sollte auf die individuellen kommunalen Belange abgestimmt werden.

Literatur

- Berntzen, L. et al. (2018). Citizens as sensors: human sensors as a smart city data source. In: *SMART 2018: The Seventh International Conference on Smart Cities, Systems, Devices and Technologies* (pp. 11–18). IARIA, 22–26 July 2018, Barcelona, Spain.
- Brackett, J. W. (1990). *Software Requirements*. Curriculum Module SEICM-19; DTIC: ADA235642. Software Engineering Institute, Carnegie Mellon University, Pittsburg.

- Bundesstadt Bonn (2022). *Webauftritt des Mängelmelders der Bundesstadt Bonn*. Retrieved Jan 30, 2022, from <https://anliegen.bonn.de/>.
- Choudhury, Z. (2012). *Reporting common street problem to relevant local authority: "FixEmUp" – a software solution for Tornio city*. Bachelor's Thesis, Kemi-Tornio University of Applied Sciences.
- Dias, G. K. A., & Wakista, G. W. (2017). Physical Civic Issues Tracking Utility. *International Journal of Research*, 5(1), 99–110. <http://doi.org/10.5281/zenodo.260304>.
- Hanse- und Universitätsstadt Rostock (2022). *Klarschiff Demonstrationsseite*. Retrieved Jan 30, 2022, from <https://demo.klarschiff-hro.de>.
- Hanse- und Universitätsstadt Rostock (2022). *Klarschiff Webauftritt*. Retrieved Jan 30, 2022, from <https://www.klarschiff-hro.de>.
- Hegner, M. (2003). *Methoden zur Evaluation von Software* (IZ-Arbeitsbericht, 29). Bonn: Informationszentrum Sozialwissenschaften. <https://nbn-resolving.org/urn:nbn:de:0168-ssoar-50730-9>.
- Kreis, H. (2022). *Mark-a-Spot Webauftritt*. Retrieved Jan 30, 2022, from <https://www.markaspot.de>.
- ISO/IEC 25010:2011-03. *Software-Engineering – Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) – Qualitätsmodell und Leitlinien*. Berlin: Beuth.
- Jackson, M., Crouch, S., & Baxter, R. (2011). *Software Evaluation: Criteria-based Assessment*. Technical Report. Software Sustainability Institute.
- MySociety (2022). *FixMyStreet Webauftritt*. Retrieved Jan 30, 2022, from <https://www.fixmystreet.org>.
- Stadt Zürich (2022). *Webauftritt des Mängelmelders der Stadt Zürich*. Retrieved Jan 30, 2022, from <https://www.zueriwieneu.ch>.
- Vaerst, T. (2017): *Mängelmelder: Anregungen zur öffentlichen Infrastruktur melden*. In: P. Patze-Diordiyuchuk, P. Renner & T. Föhr (Eds.), *Methodenhandbuch Bürgerbeteiligung – Online-Beteiligung zielgerichtet einsetzen* (Band 3, pp. 99–112). München: oekom.