

# Archiv und Repository gemeinsam gedacht

## Agile Entwicklung von Forschungsdatenservices an der Universität Münster

### 1. Einleitung

Um Forschungsdaten und Publikationen langfristig verlässlich verfügbar zu halten, müssen vielerorts bestehende Systeme durch moderne, nachhaltige Lösungen ersetzt werden. Wie viele andere wissenschaftliche Einrichtungen steht auch die Universitäts- und Landesbibliothek Münster (ULB) aktuell vor dieser Herausforderung. Ziel des hier beschriebenen Projekts<sup>1</sup> ist die Konsolidierung von mindestens drei Diensten auf einer gemeinsamen technologischen Basis: InvenioRDM, einer Open-Source-Software für Repositorien.<sup>2</sup>

An der Universität Münster wurde im September 2025 der Dienst „datasafe“<sup>3</sup> auf Basis von InvenioRDM in Betrieb genommen, ein sogenanntes „Dark Archive“ zur Aufbewahrung nicht-öffentlicher Forschungsdaten für mindestens zehn Jahre. Im Februar 2026 folgte der Dienst „datastore“, ein zentrales, FAIR-konformes Repository, das seitdem ebenfalls auf InvenioRDM basiert. Zudem soll in einem nächsten Schritt der bestehende Publikationsdienst „miami“, das heißt der Dokumentenserver der Universität Münster, durch eine InvenioRDM-Instanz ersetzt werden.

Durch diese Umstellungen sollen Wartung, Weiterentwicklung und Nutzbarkeit der Dienste nachhaltig verbessert werden. Der Praxisbericht beschreibt die zentrale Projektidee, beleuchtet organisatorische und technische Herausforderungen und stellt erprobte Lösungswege sowie Best Practices aus dem Projektalltag vor.

### 2. Ausgangssituation und Projektstruktur

Die Dienste datasafe, datastore und miami basierten bisher überwiegend auf Software, die in der ULB Münster entwickelt wurde und über Jahre gewachsen war. Sie waren jedoch mit einem hohen Wartungs- und Update-Aufwand verbunden, und das Wissen über zentrale Softwarekomponenten lag bei nur wenigen Personen. Diese Abhängigkeit erhöhte das Risiko von Ausfällen und erschwerte die langfristige Weiterentwicklung der Systeme.

Mit dem Umzug der Dienste auf eine nachhaltige, gemeinschaftsorientierte Lösung wie InvenioRDM sollen diese Defizite gezielt behoben werden. Die Open-Source-Software InvenioRDM bietet eine modulare Architektur und profitiert von einer aktiven, internationalen Community, sodass Weiter-

---

1 Dieser Beitrag bezieht sich auf den Vortrag der Autor\*innen mit dem Titel „Agile Entwicklung eines Repositoriums für die Universität Münster: Herausforderungen, Lösungen und Best Practices“ am 25.06.2025 auf dem 9. Bibliothekskongress 2025 (zugleich 113. BiblioCon) in Bremen und schließt Erkenntnisse aus der weiteren Projektentwicklung bis zum Projektabschluss mit ein.

2 Vgl. <https://inveniosoftware.org/products/rdm/>.

3 Alle Dienste der ULB Münster werden klein geschrieben.

entwicklungen, Sicherheitsupdates und neue Features langfristig gewährleistet sind. Damit schafft InvenioRDM eine technologische Basis, die die Wartung, Erweiterbarkeit und Zukunftsfähigkeit der Dienste wesentlich verbessert. InvenioRDM hat sich zudem in zahlreichen internationalen Repositorien in der Praxis bewährt und ist als technologische Basis von Zenodo.org weithin bekannt.

Der Umstieg auf InvenioRDM war zudem Anlass, neue Arbeitsweisen an der ULB Münster einzuführen: Das Projektteam startete im Rahmen der Migrationsarbeiten mit einem klassischen Scrum-Ansatz,<sup>4</sup> entwickelte daraus aber eine „Scrum-Light“-Variante, die stärker auf die Bedürfnisse des Teams zugeschnitten ist. Dieser hybride Ansatz verbindet kurze Feedbackzyklen und iterative Verbesserungen mit der Freiheit, etablierte Prozesse dort zu lockern, wo sie hinderlich waren.

## 2.1 Agile Arbeitsweise: angepasst statt dogmatisch

Die agile Projektarbeit stützte sich auf ausgewählte Elemente des Scrum-Frameworks, jedoch in reduzierter Form, um besser auf die tatsächlichen Anforderungen eingehen zu können. Regelmäßige Stand-Up-Meetings, Retrospektiven und Reviews strukturierten den Austausch und machten Fortschritte sichtbar. Auf klassische Scrum-Werkzeuge wie Story Points oder detaillierte Aufwandsschätzungen verzichtete das Team bewusst, um den Fokus stärker auf die inhaltliche Arbeit zu legen.

Zudem wurde das ursprünglich starre Rollenkonzept aufgebrochen: Die konzeptionelle Arbeit lag nicht ausschließlich bei der Product Ownerin, sondern wurde gemeinsam von Entwickler\*innen und anderen Teammitgliedern getragen. Gewissermaßen ergab sich daraus auch die Rolle eines „Product Engineers“,<sup>5</sup> der als Schnittstelle zwischen dem Developer-Team und der Product Ownerin fungierte. Diese Rollenflexibilität erwies sich als besonders effektiv in einem interdisziplinären Team mit flachen Hierarchien auf der Arbeitsebene.

## 2.2 Produktvision zwischen Kontinuität und Innovation

Eine große Herausforderung bestand in der Migration von Bestandsdiensten, deren Funktionsspektrum von den Nutzenden und Betreibenden als Referenzrahmen erwartet wird. Eine vollständige Neukonzeption oder -ausrichtung war nicht möglich, da viele Funktionen und Anforderungen des Ursprungsdienstes bereits vorgegeben waren. Ziel war es daher, eine klare Produktvision zu entwickeln, die sowohl etablierte Erwartungen erfüllt als auch zukünftige Erweiterungen zulässt. Dabei galt es, eine Balance zwischen Bewährtem und Innovation zu finden, ein Spannungsfeld, das durch kontinuierliche Nutzerbeteiligung, transparente Kommunikation und pragmatische Architekturentscheidungen moderiert werden musste.

---

4 Zur Erläuterung vgl. z.B. <https://de.wikipedia.org/wiki/Scrum>, Stand: 26.03.2026.

5 Product Engineer (o. J.): What is a Product Engineer? <https://productengineer.org/what-is-a-product-engineer>, Stand: 26.03.2026.

## 2.3 InvenioRDM: Lernkurve und Potenzial

Die Entscheidung für InvenioRDM wurde trotz einer zunächst herausfordernden Einarbeitungsphase getroffen. Die Software InvenioRDM wurde ursprünglich 2006 vom CERN, der Europäischen Organisation für Kernforschung, entwickelt und wird seitdem von einer internationalen Community unter Koordination des CERN getragen und weiterentwickelt, was eine besondere Nachhaltigkeit und Zukunftsfähigkeit gewährleistet. Seit 2024 steht mit InvenioRDM eine „schlüsselfertige“ Lösung für Forschungsdatenrepositorien zur Verfügung, die auf dem InvenioRDM-Framework basiert, aber zusätzliche Komponenten und Konfigurationen mitbringt, wie z. B. eine vollständige Implementierung des DataCite-Metadatenschemas. Aufgebaut ist InvenioRDM aber nach wie vor nach den Prinzipien eines Frameworks, also eines Programmiergerüsts, das modular erweitert werden kann. Solche Architekturen sind aufgrund ihrer hohen Flexibilität und Erweiterbarkeit in der Softwareentwicklung seit Langem stark verbreitet.

Als technische Basis von InvenioRDM werden Python (Flask) im Backend und React im Frontend eingesetzt.<sup>6</sup> Beides bietet eine solide Grundlage, die sowohl modern als auch weit verbreitet ist. Zudem erfüllt das System zentrale Anforderungen an Wartbarkeit und Anschlussfähigkeit. Besonders hervorzuheben ist, dass Anpassungen an spezifische Anforderungen mit vertretbarem Aufwand umgesetzt werden können, ohne die Updatefähigkeit des Systems grundsätzlich zu gefährden.

Die Entwickler\*innen-Community zählt inzwischen mehrere Hundert aktive Mitglieder auf vier Kontinenten. Sie arbeiten größtenteils online zusammen, koordinieren sich in regelmäßigen Videokonferenzen und auf kollaborativen Plattformen und treffen sich zusätzlich teilweise mehrmals pro Jahr auf internationalen Workshops. Die zentrale Rolle des CERN als federführende Einrichtung verleiht dem Projekt nicht nur Stabilität, sondern auch eine hohe Sichtbarkeit im internationalen Forschungsumfeld.<sup>7</sup>

## 3. Best Practices im Projektalltag

Die Entwicklung und Einführung von datasafe, dem Dienst zur Archivierung von Forschungsdaten, an der Universität Münster machten früh deutlich, wie wichtig klare Strukturen und abgestimmte Prozesse für den Projekterfolg sind. Für die Dokumentation und Kommunikation, die Qualitätssicherung, das UX-Writing oder auch für die DevOps-Prozesse<sup>8</sup> wurden daher gezielt Best Practices etabliert, die eine reibungslose Zusammenarbeit, hohe technische Stabilität und eine nutzerfreundliche Gestaltung

---

6 Flask, eine Bibliothek der Programmiersprache Python, für die serverseitige Logik im Hintergrund, und React, eine Bibliothek der Programmiersprache JavaScript, für die Benutzeroberfläche, sind weit verbreitete, quelloffene Technologien mit großen Entwickler-Communities; das erleichtert langfristig die Wartung und stellt (Sicherheit-)Updates sicher.

7 Auch die ULB Münster ist eng in diese Zusammenarbeit eingebunden und hat nach der Corona-Pandemie den ersten InvenioRDM-Hackathon in Münster ausgerichtet, bei dem Entwickler\*innen aus aller Welt vor Ort gemeinsam an der Weiterentwicklung gearbeitet haben. Vgl. dazu Heeke, Brigitte: InvenioRDM-Community vernetzt sich, in: wissen|leben, Nr. 2, 4. April 2024, <https://www.uni-muenster.de/news/view.php?cmdid=13948>, Stand: 26.03.2026.

8 UX (User Experience) meint das Nutzungserlebnis von Anwendungen und umfasst im UX-Writing die nutzerzentrierte Gestaltung von Oberflächentexten. DevOps (Development und Operations) bezeichnet die Integration von Entwicklung und Betrieb durch automatisierte und kontinuierliche Prozesse.

ermöglichten. Nach dem jeweiligen Projektabschluss wurden diese Prozesse gemeinsam evaluiert und an die Arbeitsrealität angepasst.

### 3.1 Dokumentation und Kommunikation

Bei der Projektarbeit wurde deutlich, dass Dokumentation, trotz ihres manchmal mühsamen Charakters, ein unverzichtbarer Bestandteil erfolgreicher Softwareprojekte ist – auch wenn agile Methoden wie Scrum zu Recht mahnen, Dokumentation nicht zum Selbstzweck werden zu lassen. Gerade für Kolleg\*innen aus nicht-technischen Arbeitsbereichen war es entscheidend, Projekthinhalte so aufzubereiten, dass sie verständlich, aktuell und leicht zugänglich waren. Eine klare und strukturierte Dokumentation (nicht nur auf Ebene des Software-Codes, sondern das Gesamtprodukt beschreibend) trug somit wesentlich zur erfolgreichen internen Kommunikation und zur Akzeptanz der Projektergebnisse bei.

Um den Wissensaustausch weiter zu verbessern, wurde im Projekt ein besonderes Augenmerk auf Transparenz und Nachvollziehbarkeit gelegt. Entscheidungsprozesse wurden konsequent dokumentiert, um langfristig Orientierung zu bieten und die Zusammenarbeit zwischen den verschiedenen Rollen und Disziplinen zu erleichtern.

Nach dem Launch von datasafe wurde das Projekt-Wiki auf Basis des Kollaborationstools Confluence noch einmal gründlich überarbeitet. Es dient seither als zentrale Wissensplattform und bietet eine verbesserte Grundlage für die Folgeprojekte, indem es Ergebnisse, Entscheidungen und Arbeitsstände konsolidiert und dauerhaft zugänglich macht.

Stakeholder selbst werden nicht über das Wiki eingebunden, sondern in separaten Meetings oder im Rahmen von Sprint-Reviews gezielt über Projektfortschritte, Meilensteine und Entscheidungen informiert. Diese Trennung verhindert Informationsüberflutung und ermöglicht es, Stakeholdern anschließend die für sie relevanten Ergebnisse und Planungen zu präsentieren.

### 3.2 Design und Benutzerfreundlichkeit

Für die verbesserte Gestaltung der Benutzeroberfläche wurde ein professioneller Standard etabliert: Gemeinsam mit einem UX/UI-Designer entwickelte das Projektteam ein Design-System, das eine konsistente User Experience über alle Repositorieninstanzen hinweg sicherstellt. Dieses Vorgehen ermöglichte eine enge Abstimmung zwischen Entwicklung und Gestaltung und schuf eine verlässliche visuelle Grundlage, die langfristig gepflegt und weiterentwickelt werden kann. Konzipiert wurden die Mock-ups für die zukünftige Oberfläche mit Penpot, einem webbasierten Open-Source-Tool für UI/UX-Design und Prototyping.

Neben dem visuellen Erscheinungsbild spielte auch das UX-Writing eine zentrale Rolle. Darunter versteht man die Gestaltung von kurzen, klaren und nutzerorientierten Texten innerhalb der Benutzeroberfläche, z. B. bei Fehlermeldungen, Buttons, Labels oder Platzhaltern. Ziel ist es dabei, Nutzer\*innen durch Inhalte und Prozesse zu führen, Orientierung zu geben und eine positive Nutzererfahrung zu schaffen.

Im Projekt wurden hierfür so weit wie möglich bewusst Prinzipien guter Microcopy<sup>9</sup> umgesetzt: natürliche, leicht verständliche Sprache, aktive und positive Formulierungen, Verzicht auf technischen Jargon oder kryptische Fehlercodes sowie klare Hinweise, was passiert ist und welche Schritte folgen können.

Auch optische Details wie die Gestaltung von Labels, Platzhaltern und Buttons wurden sorgfältig abgestimmt, um Redundanzen zu vermeiden und die Verständlichkeit zu erhöhen. Die Wirksamkeit dieser Ansätze wurde in regelmäßigen Usability-Tests überprüft: Rückmeldungen von Nutzer\*innen halfen, Texte und Interaktionen iterativ zu verbessern und Hürden frühzeitig zu erkennen. Gut gestaltetes UX-Writing erwies sich damit als entscheidender Bestandteil der Gesamtgestaltung: Es reduzierte Frust, vermittelte Sicherheit und trug wesentlich zur Akzeptanz der neuen Systeme bei. Die hierbei gewonnene Erfahrung wird sukzessive in den Hauptprogrammcode von InvenioRDM eingebracht und kann auf diese Weise von der Community nachgenutzt werden.

### **3.3 Zusammenarbeit der Bereiche Softwareentwicklung und Systemadministration**

Die technische Infrastruktur des Projekts basierte auf modernen DevOps-Prinzipien, die Effizienz, Transparenz und Nachhaltigkeit im Entwicklungsprozess fördern. DevOps (der Begriff setzt sich aus den Wörtern „Development“ und „Operations“ zusammen) beschreibt einen Arbeits- und Kulturansatz, der Entwicklung und Betrieb von Softwareprodukten enger verzahnt, um Software durch Zusammenarbeit, Automatisierung und kontinuierliche Prozesse zuverlässiger und effizienter zu entwickeln, zu testen und bereitzustellen.

Über GitLab CI/CD-Pipelines<sup>10</sup> wurden automatisierte Deployment-Prozesse implementiert. Dadurch ließen sich neue Softwarestände schnell, sicher und reproduzierbar auf Test- und Produktionssysteme übertragen. Ein systematisches Issue-Tracking in GitLab half dem Team, Aufgaben, Bugs und Feature-Wünsche übersichtlich zu verwalten und klar zu priorisieren. GitLab-Labels erwiesen sich dabei als besonders nützlich, um Tickets nach Themen, Dringlichkeit oder Projektphase zu strukturieren und die Zusammenarbeit zwischen Entwicklung, Design und Projektleitung zu erleichtern.

Zusätzlich kam ein strukturierter Code-Review-Prozess zum Einsatz: Alle Änderungen wurden vor dem sogenannten Merge in den GitLab-Main-Branch von Kolleg\*innen geprüft. Dies diente nicht nur der Qualitätssicherung und Fehlervermeidung, sondern auch dem Wissensaustausch innerhalb des Teams.

Da für das Projektmanagement eine Abwandlung des Scrum-Frameworks verwendet wurde, wurde das Projekt in sogenannte Sprints eingeteilt. Anfangs erhielten diese Sprints noch thematisch passende Titel. Mit der Zeit wurden diese kreativer, was die zunehmende Abkehr von strengem Scrum und die Etablierung einer flexibleren, DevOps-orientierten Arbeitsweise verdeutlicht. Im Bereich der Qua-

---

<sup>9</sup> Vgl. hierzu Yifrah, Kinneret: UX writing & microcopy, Bonn 2020.

<sup>10</sup> CI = Continuous Integration (automatisches Testen des Codes) und CD = Continuous Delivery/Deployment (automatisches Ausrollen des Codes).

litätssicherung wurden zwei komplementäre Teststrategien etabliert: Automatisierte Codetests sorgen für technische Stabilität und Fehlervermeidung auf Quellcodeebene. Gleichzeitig lieferten regelmäßige Usertests aus dem Kreis potenzieller Endnutzer\*innen wichtige Rückmeldungen, mit denen die Benutzerfreundlichkeit und Funktionalität des Systems gezielt verbessert werden konnten.

## 4. Fazit und Ausblick

Das Projekt zur agilen Entwicklung eines Archivs und eines Repositoriums an der Universität Münster zeigt, wie technische Modernisierung, agile Transformation und Open-Source-Orientierung erfolgreich kombiniert werden können. Die Erfahrungen belegen: Agilität muss kontextsensibel umgesetzt werden, pragmatisch statt dogmatisch.

Mit der Inbetriebnahme der ersten Instanzen im September 2025 und Februar 2026 steht das Projekt kurz vor seinem erfolgreichen Abschluss – und kann als Best-Practice-Beispiel für andere wissenschaftliche Einrichtungen dienen, die vor ähnlichen Herausforderungen stehen: historisch gewachsene IT-Strukturen, personelle Abhängigkeiten, begrenzte Ressourcen und der Wunsch nach FAIR-konformen, nachhaltigen und benutzerfreundlichen Repositorien.

Insbesondere die Anpassung agiler Methoden an das konkrete Arbeitsumfeld, das heißt der bewusste Verzicht auf dogmatische Frameworks und die Integration verschiedener Rollenmodelle, lassen sich gut auf andere Institutionen übertragen. Auch die Entscheidung für die aktive Beteiligung an einer internationalen Community (wie der von InvenioRDM) kann als strategischer Hebel verstanden werden: Statt Insellösungen zu pflegen, wird auf ein gemeinsam getragenes Ökosystem gesetzt, das laufend weiterentwickelt wird.

Nicht zuletzt macht das Projekt deutlich, wie wichtig Transparenz, Dokumentation und kontinuierlicher Austausch sind, sowohl innerhalb des Teams als auch mit Stakeholdern und Endnutzenden. Auch wenn technologische Entscheidungen eine zentrale Rolle spielen, sind es letztlich die Nutzenden, die über den Erfolg oder das Scheitern digitaler Infrastrukturprojekte entscheiden. Die im Projektverlauf entwickelten Prinzipien und Prozesse lassen sich deshalb nicht nur auf Repositorienprojekte übertragen, sondern bieten Orientierung für viele Digitalisierungsprojekte an Hochschulen – dort, wo Wandel sowohl technische als auch kulturelle Transformation bedeutet.

Neben der technischen Modernisierung erwies sich das Projekt auch als intensiver Lernprozess für alle Beteiligten. Die Einführung neuer Technologien, Tools und Methoden – von InvenioRDM über Dev-Ops-Praktiken bis hin zu UX-Design-Standards – erforderte eine kontinuierliche Auseinandersetzung und gezielten Wissenstransfer. Stakeholder, Entwickler\*innen, Designer und Projektleitung(en) mussten sich gleichermaßen in neue Konzepte einarbeiten und Perspektiven angleichen. Diese gemeinsame Lernkurve hat nicht nur die Projektkompetenz gestärkt, sondern auch die Basis für eine gelungene Weiterentwicklung der Dienste geschaffen.

Adienne Karsten-Welker, Universitäts- und Landesbibliothek Münster, <https://orcid.org/0000-0002-0562-7393>

Karl Krägelin, Universitäts- und Landesbibliothek Münster, <https://orcid.org/0000-0002-0189-8049>

**Zitierfähiger Link (DOI):** <https://doi.org/10.5282/o-bib/6264>

Dieses Werk steht unter der Lizenz [Creative Commons Namensnennung 4.0 International](#).