

ABSCHLUSSBERICHT

1 Allgemeine Angaben

DFG-Geschäftszeichen: HO 3987/63-1 und YA 191/16-1

Projektnummer: 675645

Titel des Projekts: OPERANDI: OCR-D Performance Optimisation and Integration – Ein Implementierungspaket der OCR-D-Software für die Massendigitalisierung

Name(n) des/r Antragstellenden: Prof. Dr. Wolfram Horstmann, übertragen an Zeki Mustafa Dogan; Prof. Dr. Ramin Yahyapour

Ehemalige Antragstellende: Prof. Dr. Wolfram Horstmann (Personenwechsel von Prof. Dr. Horstmann an Hrn. Dogan am 08.01.2024)

Dienstanschrift/en: Niedersächsische Staats- und Universitätsbibliothek, Platz der Göttinger Sieben 1, 37073 Göttingen; Gesellschaft für wissenschaftliche Datenverarbeitung mbH, Burckhardtweg 4, 37077 Göttingen

Name(n) der Mitverantwortlichen: Prof. Dr. Philipp Wieder

Name(n) der Kooperationspartner*innen: Niedersächsische Staats- und Universitätsbibliothek Göttingen (SUB), Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG)

Berichtszeitraum (gesamte Förderdauer): 01.10.2021 - 31.05.2025

2 Zusammenfassung/Summary

Deutsch: Durch DFG-geförderte Erschließungs- und Digitalisierungsprojekte stehen mittlerweile zahlreiche Digitalisate von historischen Drucken zur Verfügung, die in den „Verzeichnissen der im deutschen Sprachraum erschienenen Drucke des 16.–18. Jh.“ (VD) nachgewiesen sind. Für weitergehende Forschungen ist es entscheidend, auch die Volltexte der digitalisierten Werke gezielt durchsuchen und weiterverwenden zu können. Die Techniken der Optical-Character-Recognition (OCR) ermöglichen das massenhafte Erstellen von Volltexten. Die [„Koordinierte Förderinitiative zur Weiterentwicklung von Verfahren der Optical Character Recognition“ \(OCR-D\)](#) hat seit 2015 in drei Förderphasen Lösungen für den VD-Use-Case entwickelt. Im Rahmen der dritten Phase der Förderinitiative wurde im Implementierungsprojekt „OPERANDI: OCR-D Performance Optimisation and Integration“ ein auf OCR-D basierendes Implementierungspaket zur Massenvolltextdigitalisierung mit verbessertem Durchsatz, bei besserer Qualität der Ergebnisse entwickelt, das rechenintensive Prozessierungen in einer High Performance Computing (HPC) Umgebung ausführt und von anderen Vorhaben und Einrichtungen mit vergleichbaren Anforderungen nachgenutzt werden kann. OPERANDI hatte zwei Szenarien zu berücksichtigen: 1) Die OCR-Erzeugung für bereits digitalisierte Werke aus dem VD-Bestand im Zuge einer Massenvolltexterfassung und 2) die OCR-Erzeugung für neu zu digitalisierende Werke im Rahmen des Digitalisierungsprozesses mittels der Software [Goobi](#) und [Kitodo](#). In beiden Szenarien werden OCR-Vorhaben mit großen Textmengen unterstützt, wobei eine hohe Performanz bei guter Nutzbarkeit erreicht wird.

Englisch: Thanks to DFG-funded cataloguing and digitisation projects, numerous digitised copies of historical prints have become available. These are listed in the ‘Verzeichnisse der im deutschen Sprachraum erschienenen Drucke des 16.-18. Jh.’ (VD). For further research, it is crucial to be able to search and use the full texts of the digitised works. Optical character recognition (OCR) techniques enable the mass creation of full texts. As part of the third phase of the [‘Initiative for Optical Character Recognition Development’ \(OCR-D\)](#), the implementation project ‘OPERANDI: OCR-D Performance Optimisation and Integration’ developed an OCR-D-based implementation package for large-scale full-text digitisation, achieving improved throughput alongside higher quality results, with compute-intensive processes executed in a High Performance Computing (HPC) environment. The package can be reused by other projects and institutions with comparable requirements. OPERANDI took into account two scenarios: (1) OCR generation for already digitised works from the VD collections in the course of large-scale full-text recognition, and (2) OCR generation for newly digitised works within the digitisation workflow using the [Goobi](#) and [Kitodo](#) software. In both scenarios, OCR projects involving large text corpora are supported, combining high performance with good usability.

3 Arbeits- und Ergebnisbericht

3.1 Ausgangslage und Zielsetzung

OPERANDI setzte sich zum Ziel, eine performante, skalierbare Implementierung für die Massenvolltextdigitalisierung zu erreichen, welche in einer HPC-Umgebung zur Ausführung kommt und adaptive, parallelisierte Workflows unterstützt. Besonderes Augenmerk wurde dabei sowohl auf die Reduzierung der Prozessierungszeit als auch auf die Integration der OCR-D-Software mit den Digitalisierungsworkflow-Softwares Goobi und Kitodo gelegt. Die Projektentwicklungen entstanden in enger Abstimmung mit dem OCR-D-Koordinierungsprojekt (OCR-D-Koord) sowie den anderen Implementierungs- und Modulprojekten (IP und MP), um eine nahtlose Integration in das OCR-D-Softwarepaket zu gewährleisten.

Für Szenario 1, die OCR-Erzeugung für bereits digitalisierte Werke aus dem VD-Bestand, war es ein zentrales Erfolgskriterium, den Durchsatz gegenüber einer vor Projektbeginn durchgeführten beispielhaften Testprozessierung deutlich zu verbessern: von 0,68 OCR-Seiten pro Minute auf 30 Seiten pro Minute. Für Szenario 2, die Einbindung in Digitalisierungssoftware, wurde die Einbindung in die Workflowschritte von Goobi als Proof-of-Concept und in die architektonisch verwandte Software Kitodo als Projektziel angestrebt.

3.2 Arbeitsschritte im Berichtszeitraum

Die Arbeitsschritte im Berichtszeitraum waren in vier Arbeitspakete (AP) gegliedert: AP 1: Anforderungsmanagement, AP 2: Spezifikation und Implementierung, AP 3: Test und Evaluation der Implementierung und AP 4: Koordination und Kommunikation. Das Vorhaben verfolgte einen agilen Ansatz, bei dem die Aufgaben in mehrere kürzere Iterationen unterteilt waren und die Zwischenergebnisse in die nachfolgenden Arbeitspakete integriert wurden. Dadurch konnte flexibel auf sich ändernde Anforderungen und identifizierte Synergien reagiert werden.

3.2.1 AP 1: Anforderungsmanagement

Im Rahmen von AP 1 wurden die Anforderungen erhoben, die in AP 2 und 3 implementiert und getestet wurden. Die in einer Pilotierung erfassten initialen Anforderungen dienten als Basis für eine erste Spezifikation und Implementierung. Die Anforderungen wurden in einem agilen Product-Backlog bei [Jira](#) erfasst und verwaltet, welches mittels agiler Methoden, wie z.B. der Eisenhower-Matrix, fortlaufend angepasst und weiterentwickelt wurde.

Die für das Projekt relevanten Stakeholder wurden zu Projektbeginn durch eine Stakeholderanalyse ermittelt. Zu ihnen gehörten an vorderster Stelle OCR-D-Koord und die anderen IP und MP, zudem u. a. VD18-Bibliotheken, Pilotbibliotheken in OCR-D und das [Göttinger Digitalisierungszentrum \(GDZ\)](#). Bei bilateralen Gesprächen und regelmäßigen

Review-Meetings (s. 3.2.4) gab es Gelegenheit, Feedback zu geben und die Anforderungen mit dem Projektteam zu diskutieren. Darüber hinaus wurden bei einem Minimum Viable Product (MVP)-Workshop mittels der Storymapping-Methode die Anforderungen gruppiert und priorisiert. Das Projekt beteiligte sich zudem an allen von OCR-D-Koord organisierten Formaten, wie dem regelmäßigen *OCR-D Open Tech Call*, *OCR-D GT Call*, *OCR-D in der Praxis*, *OCR-D Forum* und den *OCR-D Workshops* mit den anderen IP und MP, wobei die Anforderungen ebenfalls aktualisiert und verfeinert wurden.

3.2.2 AP 2: Spezifikation und Implementierung

AP 2 widmete sich der iterativen Implementierung der in AP 1 ermittelten Anforderungen. Zudem stellte AP 2 die IT-Infrastruktur mit HPC-System und Storage sowie die benötigte Entwicklungsumgebung bereit und entwickelte Hilfswerkzeuge für den Workflow. Nach jedem Sprint wurden Mini-Releases mit inkrementellen Funktionen ausgeliefert, die das Produkt kontinuierlich erweiterten und verbesserten.

3.2.2.1 Workflow-Auswahl

Zu Projektbeginn wurde die in der Pilotierung begonnene Evaluierung von Workflowbeschreibungsframeworks fortgesetzt und die Workflows (WFs) [Luigi](#), [Airflow](#), [Prefect](#), [Nextflow](#) und [Flowable](#) miteinander verglichen. Als Ergebnis der Evaluierung wurde Nextflow als für das Projekt optimales WF-Framework ausgewählt, das dem im Projektantrag zunächst favorisierten WF Flowable überlegen ist. Die Gründe dafür sind, dass Nextflow

- auf wissenschaftliche Datenverarbeitung und Forschung besser zugeschnitten ist als die universell einsetzbare WF-Engine Flowable;
- mit der Beschreibung und Parametrisierung der Prozessierungsschritte über eine domänenspezifische Sprache (DSL) wiederverwendbare Teilworkflows und zusammengesetzten WFs entstehen, die in einer dynamischen Umgebung wie OCR besser anpassbar sind als mit der grafischen Oberfläche von Flowable;
- durch Unterstützung für die Containerisierung mit Docker/Apptainer plattformübergreifende Portabilität gewährleistet und damit für eine HPC-Umgebung besser geeignet ist;
- eine größere Community hat und regelmäßig aktualisiert und verbessert wird;
- flexibler bei der Skripterstellung und für Skalierung über verteilte Systeme hinweg optimiert ist.

3.2.2.2 Architektur

Die technische Umgebung der im Rahmen der Pilotierung entworfenen Architekturspezifikation wurde in Iterationen wesentlich verbessert (s. Abb. 1). Sie stellt sicher, dass sich OPERANDI in die OCR-D-Infrastruktur einfügt und in Digitalisierungssoftware integriert sowie die

Prozessierung in der HPC-Umgebung ausgeführt werden kann. Über diese initialen Anforderungen hinausgehend, wird eine sich direkt anschließende Archivierung der Ergebnisse im DFG-geförderten Projekt [“OLA-HD Service – ein generischer Dienst für die Langzeitarchivierung historischer Drucke” \(OLA-HD\)](#) angeboten. OLA-HD Service dient der Datensicherung sowie der dauerhaften und stabilen Archivierung und Referenzierung digitalisierter Inhalte und OCR-Ergebnisse. Zudem wurde im OLA-HD ein Button implementiert, um niedrigschwellig OCR-Prozessierungen über OPERANDI zu starten. Diese beiden Produkte im OCR-D-Framework wurden so nutzerfreundlich miteinander verknüpft.

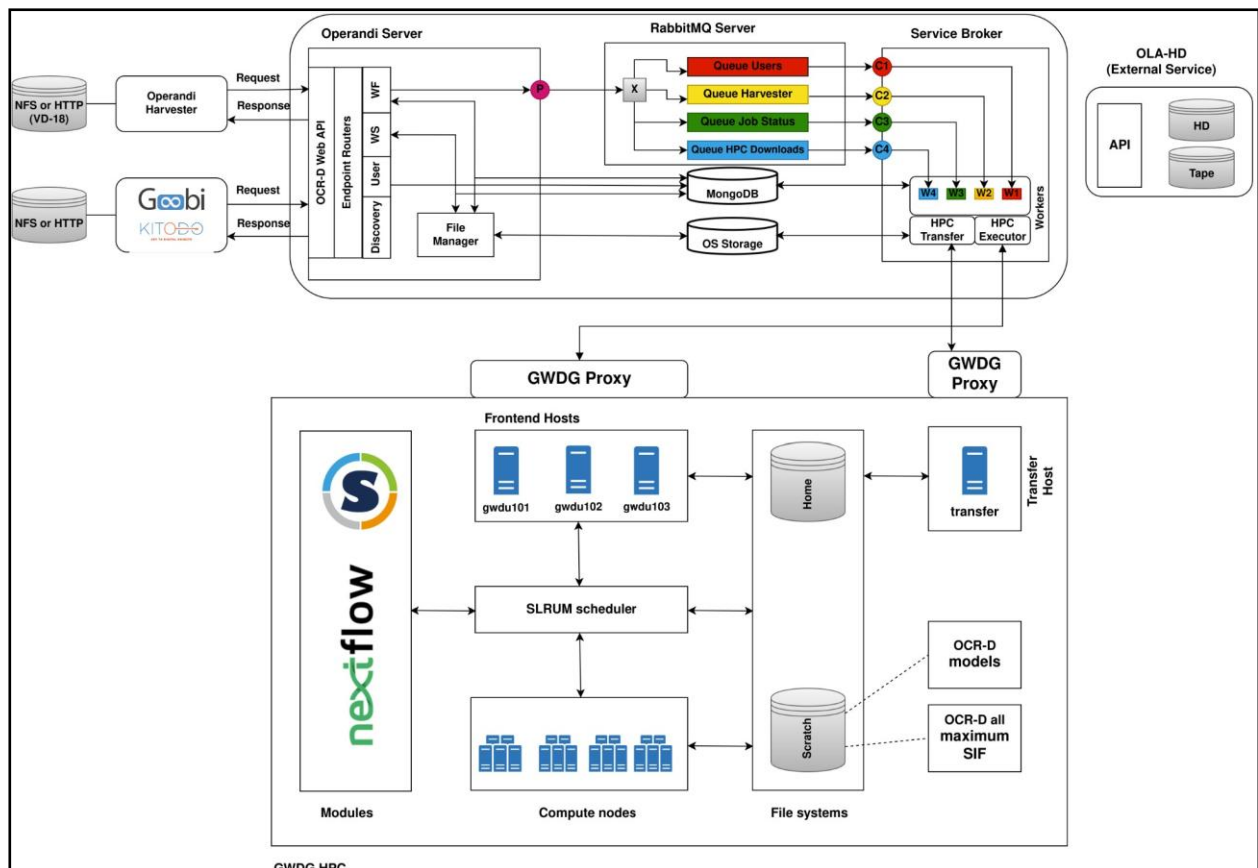


Abb. 1: OPERANDI-Architektur

Der OPERANDI-Harvester und die Goobi- bzw. Kitodo-Integration (s. 3.2.2.3) bilden die Haupteinstiegspunkte in das System. Diese Komponenten sind verantwortlich für den Bezug von Daten und das Anstoßen von Verarbeitungsworkflows.

Der OPERANDI-Server fungiert als zentrale Steuerungs- und Vermittlungseinheit, dient als Orchestrator für alle OCR-bezogenen Vorgänge und stellt eine standardisierte OCR-D Web-API bereit, über die Anfragen sowohl von Harvester-Komponenten als auch von externen Systemen entgegengenommen werden können. Nach Eingang einer Anfrage nutzt der Server interne Endpoint-Router, um Aufgaben an die entsprechenden Dienste weiterzuleiten, darunter WF-Management, Workspace Services (WS) und den File Manager. Zudem übernimmt der Server Aufgaben im Bereich Session-Management oder Job-Tracking. Er ist über Schnittstellen mit

MongoDB zur Speicherung von Job- und Nutzerdaten sowie mit dem Betriebssystemspeicher zur Zwischenablage von Dateien vor und nach der Verarbeitung verbunden. Zwecks Skalierbarkeit werden Aufträge asynchron an RabbitMQ übergeben.

Der RabbitMQ-Server bildet das Nachrichten-Backbone des Systems. Er ermöglicht eine zuverlässige und skalierbare Verteilung von Aufgaben, indem er Aufträge, die von Nutzenden oder automatisierten Harvester-Komponenten eingereicht wurden, in Warteschlangen organisiert. Die Arbeitslast wird dabei über mehrere spezialisierte Queues verteilt.

Für die eigentliche Ausführung der Aufgaben übernimmt der Service Broker die Rolle des Verarbeitungsmanagers und fungiert als Middleware, insbesondere für die Koordination mit der HPC-Infrastruktur. Er erstellt dynamisch Worker-Prozesse, die jeweils für bestimmte Aufgabenbereiche zuständig sind, wie z. B. Jobausführung oder Datenübertragungen. Rechenintensive Aufgaben werden an den HPC-Executor weitergeleitet, während Dateioperationen über das HPC-Transfer-Modul abgewickelt werden. Ähnlich wie der Server besitzt auch der Service Broker Schnittstellen zu MongoDB und zum OS-Speicher, um die Konsistenz der Daten über Komponenten hinweg sicherzustellen.

Für eine sichere Anbindung an externe Rechenressourcen nutzt das System einen GWDG-Proxy, der Authentifizierung, Autorisierung und Kommunikation gewährleistet und so Datenintegrität und Zugriffskontrolle gegenüber der GWDG-HPC-Infrastruktur sicherstellt.

Die GWDG-HPC-Umgebung bildet das Rückgrat der OCR-D-Verarbeitung. Sie besteht aus Frontend-Hosts als Zugangsknoten für die Job-Einreichung und einem SLURM-Scheduler zur Verteilung von Batch-Jobs auf zahlreiche Rechenknoten. Die WFs werden mit Nextflow orchestriert (s. 3.2.2.1). Zur Isolierung und Ausführung der OCR-D-Prozessoren kommen Apptainer-Container (zuvor Singularity) zum Einsatz. Temporäre Arbeitsdaten werden im schnellen Scratch-Dateisystem bereitgestellt, während dauerhafte Nutzerdaten im Home-Verzeichnis gespeichert werden. Der Datenverkehr in und aus dem HPC-Cluster wird über einen dedizierten Transfer Host gesteuert. Abschließend werden die verarbeiteten Ergebnisse zusammen mit den Ausgangsdaten im OLA-HD gespeichert.

3.2.2.3 Integration mit Goobi und Kitodo

Über ein Integrationsskript kann OPERANDI in Digitalisierungsplattformen wie Goobi und Kitodo ausgeführt werden. Im Projekt wurde dazu ein dynamisches Bash-Skript entwickelt, das verschiedene Digitalisierungsszenarien handhaben kann, darunter Verarbeitung von METS-basierten Daten, von Rohbilddaten oder Arbeitsbereichen im OCRD-ZIP Format. Das Skript unterstützt sowohl native OCR-D-Installationen als auch Docker-basierte Umgebungen und bietet damit Flexibilität für unterschiedliche Installationsumgebungen. Wird Kitodo in einer Docker-Umgebung ausgeführt, kann das OPERANDI-Integrationsskript deutlich einfacher und automatisch integriert werden. Damit sinkt der Aufwand für Einrichtung und Einführung spürbar.

Das Skript automatisiert die gesamte Pipeline: Erstellung des Workspaces¹, Übermittlung des WFs, Statusüberwachung, Herunterladen der Ergebnisse und Hochladen ins OLA-HD. Es umfasst zudem eine Integration des LAREX-Viewers zur visuellen Überprüfung der erzeugten Volltexte im PAGE XML Format² und optionale E-Mail-Benachrichtigungen über den Auftragsstatus mit anpassbaren SMTP-Einstellungen. Das Skript ist modular; Nutzende können Ressourcen, WFs und Dateigruppen mit Flags und Parametern anpassen. Im GitHub-Repository werden Beispielszenarien für die Verwendung des Skripts bereitgestellt.

Abb. 2 zeigt einen vereinfachten Digitalisierungsworkflow mit Kitodo (Goobi ist äquivalent) und verdeutlicht, wie OPERANDI in den Prozess eingebettet ist. Mit Goobi erfolgt die Einbettung, indem ein Schritt in der Abfolge der Aufgaben in den Vorgangsdetails ergänzt wird. Die OCR-Ergebnisse werden am Ende des WFs als neue Dateigruppe in der Kitodo- bzw. Goobi-METS-Datei integriert.

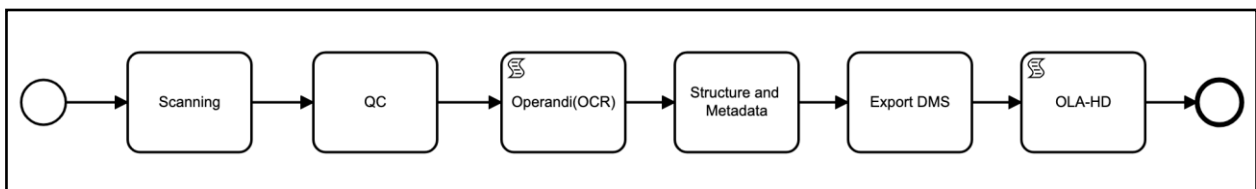


Abb. 2: Digitalisierungsworkflow mit Kitodo

3.2.3 AP 3: Test und Evaluation der Implementierung

Im AP 3 wurden durch das Ausführen von OPERANDI mit vordefinierten Datasets Fehler, Mängel oder Ursachen für nicht erreichte Erfolgskriterien identifiziert, welche dann in der nächsten Iteration berücksichtigt wurden. Abgesehen von den projektspezifischen Tests wurden die Prinzipien von Continuous Integration (CI) bzw. Continuous Delivery (CD) verfolgt und dabei zur Qualitätssicherung Unit Testing angewendet.

Vor dem Release des Minimum Viable Products (MVP) wurden Leistungskennzahlen für drei OCR-WFs mit Frakturschrift und unterschiedlicher Seitenanzahl miteinander verglichen. Die Ergebnisse verdeutlichten trotz identischer Ressourcenallokation eine nichtlineare Zunahme von Verarbeitungszeit und Speicherverbrauch bei größeren Workspaces. Die durchschnittliche Verarbeitungszeit pro Seite lag bei 25-32 Sekunden, je nach Workspace. Diese Testläufe zeigten an, dass ressourcenintensive Prozessschritte bei Skalierung gezielt optimiert werden mussten. Entsprechende Optimierungen wurden im weiteren Projektverlauf implementiert.

¹ In den Workspace werden die Bilddaten, die über die METS-Zugriffspunkte referenziert sind, geladen und zusammen mit der zugehörigen METS-Beschreibung abgelegt. Durch die Prozessierung wird der Workspace sukzessive um die Prozessierungsergebnisse erweitert.

² Eine Konversion ins ALTO XML Format ist mit OCR-D Werkzeugen möglich.

Für den Final Release wurden iterativ fünf Datensätze von durch das GDZ kuratierten VD-Texten prozessiert, um eine VD-Massenvolltextdigitalisierung zu erproben.³ Um den Durchsatz zu evaluieren und zu verbessern, wurden zwei WFs, "Standard" und "ODEM", unter einer einheitlichen Hardwarekonfiguration von 8 CPUs und 64 GB RAM pro Auftrag getestet, mit dem Ziel, den im Projektantrag angestrebten Durchsatz von 30 Seiten pro Minute zu erreichen. Einzel⁴ verarbeitete der Standard-WF lediglich 6,63 Seiten pro Minute, während der ODEM-WF mit 8,77 Seiten pro Minute etwas besser abschnitt. Die Ausführung von fünf parallelen Standard-Aufträgen oder vier parallelen ODEM-Aufträgen übertraf jedoch 30 Seiten pro Minute, was die Praktikabilität einer parallelen Skalierung bestätigt:

- 5 parallele Standard-WF-Aufträge: $5 \times 6,63 = 33,15$ Seiten/Minute
- 4 parallele ODEM-WF-Aufträge: $4 \times 8,77 = 35,08$ Seiten/Minute

Um Engpässe zu ermitteln und Optimierungen vorzunehmen, wurde jeder Workflowschritt hinsichtlich der Ausführungszeit, des Spitzen-Resident-Set-Size und des virtuellen Speichers bewertet. In der fünften Iteration wurden die WFs mit gleichen Workspaces verglichen, um eine Bewertung anhand derselben Metriken zu ermöglichen. Die Ergebnisse bestätigten, dass die Auswahl der WFs erhebliche Auswirkungen auf die Ressourceneffizienz und die Laufzeitleistung hat. Zudem ergaben weiterführende Tests eine Genauigkeit von 76% bei der Vorhersage der erforderlichen Speicherzuweisung, was hilft, "out of memory"-Fehler und eine Überbelegung von Ressourcen zu vermeiden und damit die HPC-Kosten zu reduzieren.

3.2.4 AP 4: Koordination und Kommunikation

AP 4 beschäftigte sich mit der Planung, Koordination und Steuerung der Aufgaben im Projekt sowie des Berichtswesens und der Dokumentation und fungierte als Schnittstelle zu den Stakeholdern. Es wurde ein agiler Scrum-Workflow angewendet, um Entwicklung und Implementierung zu steuern. Sprints mit regelmäßigen Meetings ermöglichten es, in kurzen Abständen Ergebnisse zu erzielen und diese gemeinsam mit den Stakeholdern zu überprüfen:

- Sprint-Planning: Vereinbarung der Anforderungen und Aufgaben zu Beginn des Sprints.
- Daily Scrum: Tägliches Treffen des Entwicklungsteams, um gemeinsam Hindernisse aus dem Weg zu räumen und Aufgaben zu planen.
- Sprint-Review: Ergebnisvorstellung und -diskussion am Ende des Sprints.
- Sprint-Retrospektive: Treffen des Scrum-Teams nach dem Review, um den Sprint zu reflektieren und Verbesserungsvorschläge für die Zusammenarbeit zu machen.

³ Ergebnisberichte bei GitHub verfügbar: <https://github.com/subugoe/operandi/blob/main/docs/Appendix.md>

⁴ "Einzel" meint, dass ein Workflow zur selben Zeit verarbeitet wird, die Prozessschritte innerhalb des Workflows werden parallelisiert abgearbeitet.

- Backlog-Refinement: Wöchentliche Überprüfung, Priorisierung und Verfeinerung der im Product-Backlog festgehaltenen Anforderungen durch das Team.

Zum virtuellen Sprint-Review wurden neben den Projektleitenden von SUB und GWDG Vertretende aus OCR-D-Koord und Projektbeteiligte aus den anderen IP und MP eingeladen. Transparenz und regelmäßige Kommunikation waren damit sowohl innerhalb des Scrum-Teams als auch im Austausch mit der OCR-D-Community gewährleistet. Ein Expertenworkshop im Februar 2023, bei dem die Beta-Version von OPERANDI vorgestellt wurde, bot eine zusätzliche Feedbackmöglichkeit und ermöglichte es Stakeholdern, ein besseres Verständnis für die vorgeschlagenen Lösungen zu gewinnen und zur Weiterentwicklung beizutragen.

Zudem wurde im Rahmen von AP 4 die Projektkommunikation koordiniert. Für die interne Kommunikation wurden Chaträume (über [RocketChat](#) und [Matrix](#)) und Videokonferenzräume (über [BigBlueButton](#)) zur Verfügung gestellt und administriert. Das Projektwiki (zunächst [Confluence](#), später [OpenProject](#)) diente nicht nur der projektinternen Abstimmung, sondern war auch für die anderen OCR-D-Projekte einsehbar und stellte so Transparenz her.

3.3 Ergebnisse

Nach Erstellung eines OPERANDI-Accounts ist OPERANDI Nutzenden zugänglich über:

- die OPERANDI Rest API: <https://operandi.ocr-d.de/docs>;
- Goobi oder Kitodo: <https://github.com/subugoe/Operandi-Integration-Script>;
- OLA-HD: <https://ola-hd.ocr-d.de/>. Für jedes Werk kann über einen Button in der Detailansicht der Default-Workflow in OPERANDI gestartet werden.

Der Code liegt Open Source bei Github: <https://github.com/subugoe/operandi>. Dort stehen unter <https://github.com/subugoe/operandi?tab=readme-ov-file#readme> auch Nutzerdokumentation und Beispielnutzungsszenarien zur Verfügung.

3.4 Öffentlichkeitsarbeit

Die Öffentlichkeitsarbeit im Rahmen von OPERANDI stellte ein Instrument zur Schaffung eines Resonanz- und Reflektionsraums für die Projektarbeit dar und konzentrierte sich vor allem auf den Austausch und die Diskussion von Fragestellungen, Anforderungen und möglichen Lösungen innerhalb der OCR-D-Community:

- Präsenz auf der [OCR-D-Website](#);
- quelloffene Bereitstellung des Softwarecodes und der technischen Dokumentation via GitHub (s. 3.3);
- regelmäßige Teilnahme an den von OCR-D-Koord organisierten Veranstaltungsformaten und an den insgesamt fünf Workshops für IP und MP;
- Expertengespräche zur Anforderungsanalyse und zum Projektfortschritt;

- Teilnahme am DFG-geförderten VD-Rundgespräch im August 2023;
- weitere Vermittlung der Erfahrungen und Ergebnisse in geeigneten Gremien (NFDI, Text+, Kitodo e.V., DINI e.V.).

4 Öffentlich zugängliche Projektergebnisse

4.1 Publikationen mit wissenschaftlicher Qualitätssicherung

4.2 Weitere Publikationen und öffentlich gemachte Ergebnisse

- Baierer, Konstantin; Boenig, Matthias; Hinrichsen, Lena; Mustafa, Mehmed; Reul, Christian; Sautter, Lilja; Will, Larissa: Das richtige Tool für die Volltextdigitalisierung. Workshop auf der Dhd2024, Passau, 27.02.2024.
- Baierer, Konstantin; Mustafa, Mehmed: OPERANDI and ocrd-k8s: Scaling up OCR-D Workflows. Dhd AG OCR Workshop, Mannheim, 27.-28.11.2024.
- Dogan, Mustafa; Gottschling, Maxie; Liewald, Kay; Pfennigstorf, Ingo; Raschkowski, Rafael Maria; Sautter, Lilja; Schima-Voigt, Kristine: Agile Research Software Development: Scrum. Vortrag bei der deRSE23 - Conference for Research Software Engineering in Germany, Paderborn, 21.02.2023. <<https://doi.org/10.5281/zenodo.7715041>>
- Doan, Triet; Panzer, Jörg; Sautter, Lilja: OPERANDI – OCR-D Performanzoptimierung und Integration. Vortrag beim Kitodo-Anwendertreffen, 25.11.2021. <https://www.kitodo.org/fileadmin/groups/kitodo/Dokumente/Anwendertreffen_2021_Operandi.pdf>
- Hartwig, Uwe; Hinrichsen, Lena; Marcus, Sven; Mustafa, Mehmed; Reul, Christian; Sautter, Lilja; Sommer, Erik; Weigelt, Markus; Weil, Stefan; Rykhinskaya, Ekaterina: OCR für die Massendigitalisierung: Einführung in OCR-D. Hands-on-Lab bei der BiblioCon 2023, Hannover, 24.05.2023.
- OPERANDI GitHub-Repository: <<https://github.com/subugoe/operandi>>