

Abschlussprüfung Sommer 2025

Fachinformatiker fr. Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

**Abgabetermin: 26. Mai 2025**

Entwicklung eines Odoo-Moduls zur

**ZIP-komprimierten Dateiverwaltung** im

**Dokumentenmanagementsystem** (DMS)

**Diese Dokumentation wurde erstellt von:**

Mark Albers

Tienrade 1, 21031 Hamburg

E-Mail: [albers.mark02@gmail.com](mailto:albers.mark02@gmail.com)

Prüflingsnummer: (131)- 54068

**Ausbildungsbetrieb:**

BFW-Berufsförderungswerk Hamburg GmbH

Marie-Bautz-Weg 16, 22159 Hamburg

Ansprechpartner: Dr. Olaf Kubillus

**Betrieb für die Praktische Ausbildung:**

Sirum GmbH

Am Sandtorkai 32, 20457 Hamburg

Ansprechpartner: Dipl. Ing. (FH) Georg Notter

E-Mail: [Georg.Notter@Sirum.de](mailto:Georg.Notter@Sirum.de)

Inhalsverzeichnis

[1. Einleitung - 1 -](#_Toc198052179)

[1.1. Projektumfeld - 1 -](#_Toc198052180)

[1.2. Projektbeschreibung - 1 -](#_Toc198052181)

[1.3. Projektziel - 1 -](#_Toc198052182)

[1.4. Projektbegründung - 1 -](#_Toc198052183)

[1.5. Projektschnittstellen - 2 -](#_Toc198052184)

[2. Projektplanung - 2 -](#_Toc198052185)

[2.1. Projektphasen - 2 -](#_Toc198052186)

[2.2. Ressourcenplanung - 2 -](#_Toc198052187)

[2.3. Entwicklungsprozess - 3 -](#_Toc198052188)

[3. Analysephase - 3 -](#_Toc198052189)

[3.1. Ist-Analyse - 3 -](#_Toc198052190)

[4. Entwurfsphase - 4 -](#_Toc198052191)

[4.1. Maßnahmen zur Qualitätssicherung - 4 -](#_Toc198052192)

[5. Kosten/Nutzen-Analyse - 5 -](#_Toc198052193)

[6. Risiko-Berechnung - 6 -](#_Toc198052194)

[7. SMART-Ziele - 7 -](#_Toc198052195)

[8. SWOT-Analyse - 8 -](#_Toc198052196)

[9. Durchführung - 9 -](#_Toc198052197)

[10. Qualitätssicherung - 10 -](#_Toc198052198)

[11. Soll-Ist-Vergleich - 11 -](#_Toc198052199)

[12. Fazit - 12 -](#_Toc198052200)

[13. Glossar - 13 -](#_Toc198052201)

[14. Anhang - 14 -](#_Toc198052202)

# Einleitung

In der folgenden Projektdokumentation wird der Ablauf des IHK-Abschlussprojektes, das der Autor im Rahmen seiner Ausbildung zum Fachinformatiker für Anwendungsentwicklung durchgeführt hat, erläutert.

## Projektumfeld

Der Ausbildungsbetrieb ist die Sirum GmbH. Sirum hat sich zum Ziel gesetzt, eine innovative Softwarelösung für die Transportlogistik zu entwickeln. Zurzeit beschäftigt Sirum etwa 30 Mitarbeiter. Während herkömmliche Lösungen oft nur einzelne Aspekte der Logistik abdecken, fokussiert sich Sirum auf eine ganzheitliche Optimierung, um Einschränkungen in der Gesamteffizienz zu überwinden. Mit dem Sirum Transport Management System (TMS), wurde eine umfassende Softwarelösung entwickelt, die auf dem Odoo-Framework basiert. Sirum setzt auf eine Software as a Service (SaaS)-Architektur, die eine transparente und kalkulierbare Kostenstruktur ermöglicht.

## Projektbeschreibung

Im Bereich der Dokumentenverwaltung wird das bestehende Document Management System (DMS) verwendet, in den verschiedenen Dateitypen wie Rechnungen, Excel-Tabellen und weitere geschäftsrelevante Dokumente abgelegt und verwaltet werden.

Das Projekt basiert auf einer konkreten Anforderung eines Kunden. Ziel war es zunächst, ein Modul zu entwickeln, mit dem ausgewählte Dateien aus dem bestehenden ir.attachment-Modell komprimiert und als ZIP-Datei heruntergeladen werden können.

Im weiteren Verlauf äußerte der Kunde zusätzliche Anforderungen: Es sollte eine benutzerfreundliche Oberfläche geschaffen werden, die eine gezielte Filterung von Dokumenten nach Datum (von–bis) und Kategorie ermöglicht. Die Ergebnisse sollten dabei in einer eigenen Benutzeroberfläche (View) angezeigt und als Datensatz gespeichert werden. Dies erlaubt es dem Nutzer, eine bestimmte Dokumentenmenge einmal zu definieren und anschließend bei Bedarf herunterzuladen, ohne den Filterprozess erneut durchlaufen zu müssen.

Das entwickelte Modul stellt somit eine umfassende Erweiterung des bisherigen DMS dar. Es unterstützt die Anwender dabei, effizienter mit großen Mengen an Dokumenten zu arbeiten und relevante Dateien gezielt zusammenzustellen und zu exportieren.

## Projektschnittstellen

Das entwickelte Modul greift auf das bestehende Modell ir.attachment zu, in dem alle im DMS gespeicherten Dokumente verwaltet werden. Es erweitert die Funktionalität dieses Modells um Filter-, Zwischenspeicherungs- und Downloadoptionen. Die Dokumente selbst bleiben im System bestehen und werden nicht verändert, sondern lediglich referenziert, gefiltert und für den Download vorbereitet.

Eine weitere Schnittstelle besteht zur View, über die der Anwender mit dem Modul interagiert. Diese View wurde speziell für das Projekt neu erstellt und nutzt einen Wizard zur Eingabe der gewünschten Filterkriterien (Datum von–bis, Kategorie). Die gefilterten Ergebnisse werden anschließend in einem temporären Record gespeichert, um eine wiederholte Verwendung zu ermöglichen.

## Projektbegründung

Im Arbeitsalltag der Fachabteilungen werden regelmäßig große Mengen an Dokumenten im DMS verwaltet, darunter unter anderem Rechnungen, Excel-Dateien und Verträge. Bisher fehlte jedoch eine Möglichkeit, diese Dokumente gezielt zu filtern und gesammelt herunterzuladen. Stattdessen mussten die benötigten Dateien manuell gesucht, einzeln ausgewählt und heruntergeladen werden, was ein zeitaufwändiger und fehleranfälliger Prozess ist.

Das Modul leistet einen Beitrag zur Standardisierung der Prozesse, indem es eine strukturierte und wiederverwendbare Lösung innerhalb des bestehenden Systems bietet. So wird nicht nur der manuelle Aufwand reduziert, sondern auch die Fehleranfälligkeit im Umgang mit Dokumenten verringert.

## Projektziel

Das Ziel des Projekts ist die Entwicklung eines Moduls zur effizienten Verwaltung und gezielten Bereitstellung von Dokumenten innerhalb des bestehenden DMS. Die Anwendung soll es den Benutzern ermöglichen, Dokumente anhand eines definierten Datumsbereichs sowie bestimmter Kategorien zu filtern, diese in einer übersichtlichen View darzustellen und bei Bedarf gebündelt als ZIP-Datei herunterzuladen.

# Projektplanung

In diesem Kapitel geht es darum, wie das weitere Vorgehen bestimmt wurde. Das Projekt wurde dazu zeitlich geplant, alle benötigten Ressourcen ermittelt und der Entwicklungsprozess bestimmt.

## Projektphasen

Für die Umsetzung des Projekts standen 80 Stunden zur Verfügung. Im Rahmen der Projektplanung wurden die Stunden auf verschiedene Phasen aufgeteilt und somit der gesamte Projektablauf abgebildet. Eine grobe Zeitplanung mit den Hauptphasen kann aus Tabelle 1 entnommen werden. Eine detaillierte Zeitplanung mit den einzelnen Schritten der jeweiligen Phase befindet sich im Anhang [**A.1: Detaillierte Zeitplanung**](#_Anhang) auf Seite i.

|  |  |
| --- | --- |
| Projektphase | Geplante Zeit |
| Analysephase | 0 h |
| Entwurfsphase | 0 h |
| Implementierung inkl. Tests | 0 h |
| Abnahme und Einführung | 0 h |
| Dokumentation | 0 h |
| Gesamt | 0 h |

Tabelle 1: Zeitplanung

## Ressourcenplanung

Im Anschluss an die Zeitplanung wurden alle für die Umsetzung des Projekts benötigten Ressourcen im [**Anhang A.2: Ressourcenplanung**](#_Anhang) aufgelistet. Dabei wurden sowohl die eingesetzten Software- und Infrastrukturressourcen als auch das benötigte Personal berücksichtigt.

Für die Entwicklung kamen ausschließlich bereits lizenzierte oder intern bereitgestellte Werkzeuge und Umgebungen zum Einsatz. Ziel war es, die Projektkosten so gering wie möglich zu halten und keine zusätzlichen Lizenzen zu beschaffen.

Im Vorfeld wurden alternative Module mit ähnlicher Funktionalität untersucht. Aufgrund Kosten Ersparnis und Lernzwecken wurde bewusst auf deren Einsatz verzichtet. Stattdessen wurde ein eigenes Modul entwickelt, das optimal auf die spezifischen Anforderungen zugeschnitten ist und sich nahtlos in die bestehende Umgebung integriert.

## Entwicklungsprozess

Bei der Firma Sirum wird ein Kanban-basierter Entwicklungsprozess verwendet, der über das GitLab-Issue-Board organisiert ist. Das Board ist in die typischen Statusspalten **To Do**, **In Progress**, **Review** und **Done** unterteilt. Jedes neue Issue wird zunächst im Board erfasst und priorisiert.

Für jedes angelegte Issue wird ein eigener Git-Branch erstellt, auf dem die Umsetzung erfolgt. Nach der Entwicklung des jeweiligen Features werden Unit-Tests geschrieben, um die Funktionsweise des Moduls automatisiert zu überprüfen.

Anschließend erfolgt ein Code-Review durch einen Entwickler, der den Branch gemeinsam mit dem zugehörigen Issue prüft. Wenn keine Änderungen mehr notwendig sind, wird der Branch durch den unternehmensinternen Merge-Bot verarbeitet. Die dabei durchlaufenen Prüfschritte und Qualitätssicherungsmaßnahmen werden in Abschnitt [4: Maßnahmen zur Qualitätssicherung](#_Maßnahmen_zur_Qualitätssicherung) näher beschrieben.

Durch dieses strukturierte Vorgehen werden Fehler frühzeitig erkannt und können bereits während der Implementierungsphase behoben werden. Gleichzeitig wird die Qualität des Quellcodes sichergestellt und der Entwicklungsprozess bleibt jederzeit transparent und nachvollziehbar.

# Analysephase

Um zu ermitteln, was sich die Fachabteilung wünscht und wie der derzeitige Stand ist, wurde eine Analyse durchgeführt. Neben der Ermittlung der Anwendungsfälle wurde außerdem eine wirtschaftliche Betrachtung des Projektes durchgeführt.

## Ist-Analyse

Im aktuellen Zustand werden im DMS sämtliche Dateien zentral im Modell ir.attachment gespeichert. Dieses Modell dient als generische Ablage für alle Arten von Dokumenten, beispielsweise Rechnungen, Lieferscheine oder Excel-Dateien. Eine gezielte Filterung nach Kriterien wie Datum oder Kategorie ist im Standard nicht möglich. Die Suche und Auswahl von Dokumenten muss manuell über die Benutzeroberfläche erfolgen.

Ein weiterer Nachteil besteht darin, dass mehrere Dateien nur einzeln heruntergeladen werden können. Eine Funktion zum Sammel-Download mehrerer Dokumente, etwa durch eine ZIP-Komprimierung, ist nicht vorhanden. Gerade bei größeren Datenmengen stellt dies für den Benutzer einen hohen manuellen Aufwand dar. Zusätzlich limitieren moderne Webbrowser aus Sicherheitsgründen die Anzahl gleichzeitiger Downloads, was den Prozess weiter verlangsamt.

Die abgelegten Dokumente enthalten teilweise geschäftsrelevante und buchhaltungsbezogene Informationen. Eine strukturierte Verwaltung oder Trennung nach Dokumententypen ist aktuell nicht gegeben. Auch eine Möglichkeit, ausgewählte Dokumente temporär zu speichern, um sie später erneut herunterzuladen, existiert nicht.

Insgesamt ist die aktuelle Lösung funktional zwar ausreichend für die einfache Dokumentenablage, bietet jedoch keine effizienten Werkzeuge zur gezielten Auswahl, Verwaltung und strukturierten Bereitstellung von Dokumenten.

# Entwurfsphase

## Maßnahmen zur Qualitätssicherung

Zur Sicherstellung der Qualität des entwickelten Moduls wurden verschiedene Maßnahmen und Prüfprozesse durchgeführt.

Bereits während der Entwicklung wurden für zentrale Funktionen Unit-Tests erstellt. Diese Tests wurden lokal ausgeführt, um die Funktionsfähigkeit einzelner Komponenten frühzeitig zu validieren. Nach Abschluss der Entwicklung wurden die Tests erneut ausgeführt, um sicherzustellen, dass alle Funktionen wie erwartet arbeiten.

Bevor ein Feature in den Hauptzweig (Master) übernommen werden konnte, wurde für jedes GitLab-Issue ein Code-Review durch einen Entwickler durchgeführt. Dabei wurde der Quellcode geprüft und ggf. mit Kommentaren versehen, falls Verbesserungsbedarf bestand. Wenn dies erfolgreich geprüft wurde, dann musste es einen mehrstufigen Qualitätssicherungsprozess auf GitLab durchlaufen. Dieser bestand aus folgenden automatisierten Tests:

* **Deploy Test:** Prüfung, ob das Modul erfolgreich gebaut und gestartet werden kann
* **Translate Test:** Kontrolle auf korrekte Übersetzungsdateien und -strukturen
* **Lint Test:** Prüfung des Quellcodes auf Einhaltung von Syntax- und Stilvorgaben
* **Manual Action Test:** Ausführung definierter manueller Testschritte

Erst nach erfolgreicher Prüfung und Bestehen aller Tests konnte der Merge über den unternehmensinternen Merge-Bot erfolgen.

Dieser strukturierte Ablauf ermöglichte es, Fehler frühzeitig zu erkennen, die Codequalität konstant hochzuhalten und gleichzeitig einen kontrollierten Übergang der Features in das produktive System sicherzustellen.

# Kosten/Nutzen-Analyse

# Risiko-Berechnung

# SMART-Ziele

# SWOT-Analyse

# Durchführung

# Qualitätssicherung

# Soll-Ist-Vergleich

# Fazit

# Glossar

# Anhang