



Portfolio – Phase 3

IU Internationale Hochschule Fernstudium

Studiengang: Master of Science Informatik

Kurs: Projekt – Software Engineering (DLMCSPSE01_D)

Abstract

Windows Anwendung:

Priorisierungsmatrix nach dem Sung-Diagramm

Gregor Hubmann

Matrikelnummer: 9196398

Hornstraße 185

5423 St. Koloman

Abgabedatum (Phase 3): 08.12.2024

Änderungen nach Phase 2:

In Phase 3 neu erstellt.

Inhaltsverzeichnis

1. Zusammenfassung der Projektdurchführung.....	1
2. Was funktionierte gut?	1
3. Herausforderungen und unerwartete Probleme.....	1
4. Was könnte man besser machen?	2
5. Wichtige Erkenntnisse.....	2

1. Zusammenfassung der Projektdurchführung

Das Projekt zielte darauf ab, eine benutzerfreundliche Priorisierungsmatrix-Anwendung basierend auf dem Sung-Diagramm zu entwickeln. Es folgte einer klar strukturierten Methodik, die in drei Hauptphasen unterteilt war. In der Konzeptionsphase wurde die Grundlage des Projekts gelegt: eine detaillierte Projektdokumentation wurde begonnen, und sowohl das Anforderungs- als auch das Spezifikationsdokument erstellt. Hierbei stand die genaue Definition der funktionalen und nicht-funktionalen Anforderungen im Vordergrund. Diese Phase bildete das Fundament für alle weiteren Aktivitäten und wurde fristgerecht abgeschlossen (Ende Phase 1: 27.10.2024).

In der Umsetzungsphase lag der Fokus auf der technischen Realisierung. Das Architekturdokument wurde erstellt, um die Struktur der Anwendung zu definieren. Hierbei zeigte sich, dass die Balance zwischen Detailliertheit und Übersichtlichkeit nicht immer leicht zu finden war. Trotz dieser Herausforderungen gelang es, die Kernfunktionalitäten, wie Aufgabenverwaltung, Drag-and-Drop und Filtermöglichkeiten, termingerecht umzusetzen. Einige geplante Komfortfunktionen, wie die Bearbeitung von Aufgaben per Doppelklick oder eine intuitive Datumsauswahl über einen Calendar-Picker, wurden aufgrund von Zeitmangel jedoch zurückgestellt (Ende Phase 2: 24.11.2024).

Die abschließende Test- und Optimierungsphase diente der Qualitätssicherung und Verfeinerung der Anwendung. Ein umfangreiches Testdokument wurde erstellt, das sowohl automatisierte Unit-Tests als auch manuelle Systemtests umfasste. Parallel dazu wurden bestehende Funktionen optimiert und offene Punkte bearbeitet. Abschließend wurde das Projekt in einer kritischen Reflexion betrachtet, um „Lessons Learned“ für zukünftige Vorhaben festzuhalten. Die ursprünglich anvisierten Deadlines konnten in allen drei Phasen eingehalten werden, was den Erfolg des Projektmanagements unterstreicht. Hierbei gebührt auch dem ausführlichen und zeitnahen Feedback Dank, das maßgeblich zur Qualität und Zielerreichung beigetragen hat (Ende Phase 3: 08.12.2024).

2. Was funktionierte gut?

Der strukturierte Aufbau des Projekts in drei Phasen mit jeweils klar definierten Outputs erwies sich als besonders hilfreich. Dieser Ansatz sorgte für klare Orientierungspunkte und half dabei, den Fortschritt kontinuierlich zu überprüfen und zu bewerten. Ergänzt durch regelmäßige Tests und zeitnahes Feedback konnte eine angemessene Qualität der Arbeit sichergestellt werden.

3. Herausforderungen und unerwartete Probleme

Durch die begrenzte praktische Programmiererfahrung und das erstmalige eigenständige Durchführen eines solchen Projekts waren einige Aspekte besonders herausfordernd. Die GUI-Entwicklung stellte sich, insbesondere bei der Implementierung der Drag-and-Drop-Funktion, als komplexer heraus als zunächst angenommen. Hier war es schwierig, sowohl die technische Umsetzung als auch die Nutzerfreundlichkeit in Einklang zu bringen.

Auch das Zeitmanagement erwies sich als problematisch. Dies lag einerseits an der Unerfahrenheit, realistische Zeitbudgets für verschiedene Phasen einzuplanen, andererseits an parallellaufenden Verpflichtungen, die die verfügbare Arbeitszeit stark einschränkten.

Zudem nahmen auftauchende Fehler durch Änderungen oft unerwartet viel Zeit in Anspruch. Diese Debugging-Phasen zeigten deutlich, wie wichtig ein strukturiertes Testen und eine präzise Planung bei Änderungen sind, um Folgeprobleme zu minimieren.

4. Was könnte man besser machen?

Ein zentraler Punkt, der verbessert werden könnte, betrifft die verfügbare Zeit für die Projektphasen. Besonders Phase 2, in der die Architektur definiert und die Implementierung begann, und Phase 3, die den finalen Feinschliff und Tests umfasste, hätten mehr Zeit benötigt, um alle Anforderungen in der gewünschten Qualität umzusetzen. Insbesondere Phase 3, mit ihrer Vielzahl an Aufgaben wie Testing, Fehlerbehebung und Dokumentation, war zeitlich zu knapp bemessen.

Die technische Unerfahrenheit war ein weiterer Faktor, der den Ablauf beeinflusst hat. Dies zeigte sich nicht nur bei der GUI-Entwicklung, insbesondere der Drag-and-Drop-Funktion, sondern auch bei der Umsetzung der Schichtenarchitektur. So wurde in manchen Bereichen die Trennung nicht konsequent eingehalten, und Teile der Logik fanden ihren Weg in den GUIController, was ursprünglich vermieden werden sollte. Mit mehr Erfahrung im Vorfeld hätte hier konsequenter strukturiert werden können.

Auch die Dokumentation, speziell das Architekturdokument, hätte besser ausbalanciert werden können. Es war in Teilen (wie den Hauptkomponenten) zu detailliert und in anderen (z. B. beim UML-Klassendiagramm) zu oberflächlich gehalten. Dies zeigt, dass ein besseres Verständnis der erforderlichen Detailebenen und der Zweckmäßigkeit solcher Dokumente eine wichtige Verbesserung darstellt.

5. Wichtige Erkenntnisse

Das Projekt hat verdeutlicht, wie entscheidend eine realistische Zeitplanung ist. Die strukturierte Herangehensweise, unterteilt in klar definierte Phasen mit spezifischen Outputs, erwies sich als von großem Vorteil, da sie Orientierung und Fokus schaffte. Dennoch machten unterschätzte Aufgaben und unerwartete Probleme, wie komplexe Fehlerbehebungen, deutliche Pufferzeiten notwendig.

Auch die Bedeutung einer sauberen Schichtenarchitektur wurde klar, da fehlende Trennung von Logik und Präsentation zu erhöhtem Anpassungsaufwand führte. Zudem zeigte sich, dass eine iterative Herangehensweise oft effektiver ist als Perfektionismus von Beginn an. Der pragmatische Umgang mit Problemen war eine wertvolle Erfahrung, die in zukünftigen Projekten weiter vertieft werden kann.