

IU Internationale Hochschule

Weiterbildung: "Software Engineering - Python"

Modul : DLMCSPSE01_D - Projekt: Software Engineering

Tutor: Prof. Dr. David Kuhlen



Prüfungsleistung: Portfolio

Projektdokumentation

Eingereicht am 11.06.2025.

Verfasser:

Djahan Bayrami Latran

Denninger Straße 198

81927 München

E-Mail: djahan.latran@gmail.com

Matrikelnummer: UPS10672478

1. Projektübersicht

Die Idee zum Projekt entstand durch Bearbeitung des Moduls "Algorithmik", bei dem im Skript unter anderem die Vorgehensweise diverser gängiger Algorithmen (bspw. Bubble-Sort, Selection-Sort, Depth-First-Search, usw.) beschrieben wurde. Zwar gab es Tabellen zur besseren Veranschaulichung der Konzepte, jedoch waren diese zum Teil trotzdem schwer nachzuvollziehen.

Ziel des Projekts ist es, eine unter Windows lauffähige Applikation mit GUI zu entwickeln, die durch grafische Veranschaulichung in Form von Animationen und Interaktivität zu einem besseren und schnelleren Verständnis der Abläufe verschiedener Algorithmen beiträgt.

Der Nutzer soll in der Lage sein, aus einer Liste von Algorithmen zu wählen, deren Ablauf visuell zu beobachten und selbst Parameter zu verändern, wie beispielsweise die Größe eines zu sortierenden Arrays oder die Geschwindigkeit der Laufzeit. Abhängig vom jeweiligen Algorithmus soll immer die passendste Darstellung zum bestmöglichen Verständnis der Abläufe gewählt werden. Das angestrebte Ergebnis soll eine GUI-Anwendung zu Lernzwecken sein, die dem Nutzer ein solides Wissen über die Funktionsweise diverser Algorithmen durch grafische Darstellung vermitteln soll.

2. Risikomanagement

Technische Risiken:

- **Logikfehler in Algorithmen:** Die Algorithmen könnten Logikfehler in der Implementierung enthalten, weshalb es wichtig ist, frühzeitige Tests durchzuführen, um die Richtigkeit zu gewährleisten.

Eintrittswahrscheinlichkeit: Mittel

Schadensausmaß: Sehr hoch

- **Performanceprobleme der Algorithmen:** Bei der Implementierung der Algorithmen sollte nicht nur die Richtigkeit getestet werden, sondern auch die Performance bei großen Eingabemengen. Dabei sollte darauf geachtet werden, effiziente Datenstrukturen zu verwenden, damit die Applikation flüssig läuft.

Eintrittswahrscheinlichkeit: Mittel

Schadensausmaß: Hoch

- **Fehlerbehandlung mangelhaft:** Durch falsche Eingaben könnte die grafische Darstellung nicht funktionieren oder abstürzen. Simple Fehler wie beispielsweise Indexfehler oder andere ungültige Eingaben würden die Applikation zwecklos machen. Um dem entgegenzuwirken, sollte eine saubere und umfangreiche Fehlerbehandlung und Fehlermeldungen eingebaut werden.

Eintrittswahrscheinlichkeit: Niedrig (da Benutzer keine individuellen direkten Eingaben tätigen kann)

Schadensausmaß: Sehr hoch

- **Speicher/Ressourcenverbrauch der grafischen Darstellung:** Es besteht das Risiko, dass durch die grafische Darstellung und ihrer Frame-basierten Aktualisierung zu viel Speicher und Ressourcen verwendet werden. Dies könnte zu unerwarteten Abstürzen der Applikation führen. Es sollte darauf geachtet werden, eine passende und etablierte Bibliothek oder Framework für die grafische Darstellung zu verwenden, deren genutzte Datenstrukturen auf diesen Use-Case optimiert sind.

Eintrittswahrscheinlichkeit: Niedrig

Schadensausmaß: Mittel

- **Probleme bei Integration eines Fensters mit grafischen Animationen in ein GUI:** Die Integration eines Fensters, das Grafiken generiert in ein GUI, könnte potentielle Probleme auslösen. Hier müsste recherchiert werden, ob für die beiden Funktionen unterschiedliche Threads verwendet werden sollten, damit diese unabhängig voneinander laufen. Dadurch könnte gewährleistet werden, dass das GUI reaktionsfähig bleibt, auch wenn das Grafikfenster ein Problem hat. Bei Verwendung von unterschiedlichen Bibliotheken für die Funktionen sollte auf die Kompatibilität dieser geachtet werden.

Eintrittswahrscheinlichkeit: Hoch

Schadensausmaß: Sehr Hoch

Management Risiken:

- **Unklarheiten in der Anforderung:** Da die Applikation theoretisch unbestimmt viele Darstellungsmöglichkeiten und Algorithmen beinhalten könnte, besteht das Risiko, dass das Projekt unkontrolliert wächst oder zu komplex wird. Um dem entgegenzuwirken, sollte der Umfang von Anfang an klar definiert im Anforderungsdokument festgelegt werden.

Eintrittswahrscheinlichkeit: Hoch

Schadensausmaß: Hoch

- **Nutzer ignorieren:** Die Anwendung könnte ihren Sinn und Zweck verfehlen, wenn nicht auch Feedback von den zukünftigen Nutzern (in diesem Fall dem Tutor) während der Entwicklungsphase eingeholt wird. Dadurch wird sichergestellt, dass die Anwendung eine Nutzergruppe anspricht und deren Anforderungen erfüllt.

Eintrittswahrscheinlichkeit: Gering

Schadensausmaß: Hoch

- **Unrealistische Zeitplanung/Auftreten von unerwarteten Problemen:** Der Aufwand zur Umsetzung der geplanten Idee könnte unterschätzt werden. Erfahrungsgemäß können sich außerdem unerwartete Probleme während der Bearbeitung ergeben, die in der Kalkulation der Zeitplanung nicht berücksichtigt wurden. Aus diesem Grund sollte in die Konzeptionsphase besser zu viel als zu wenig Zeit investiert werden

Eintrittswahrscheinlichkeit: Hoch

Schadensausmaß: Mittel

- **Zeitmangel durch nebenberufliche Weiterbildung:** Da das Projekt nebenberuflich durchgeführt wird, kann es zu zeitlichen Verzögerungen, verursacht durch den Hauptberuf kommen. Durch eine ausführliche Vorplanung und nachholen der Stunden an Wochenenden wird versucht, dies bestmöglich zu verhindern.

Eintrittswahrscheinlichkeit: Sehr hoch

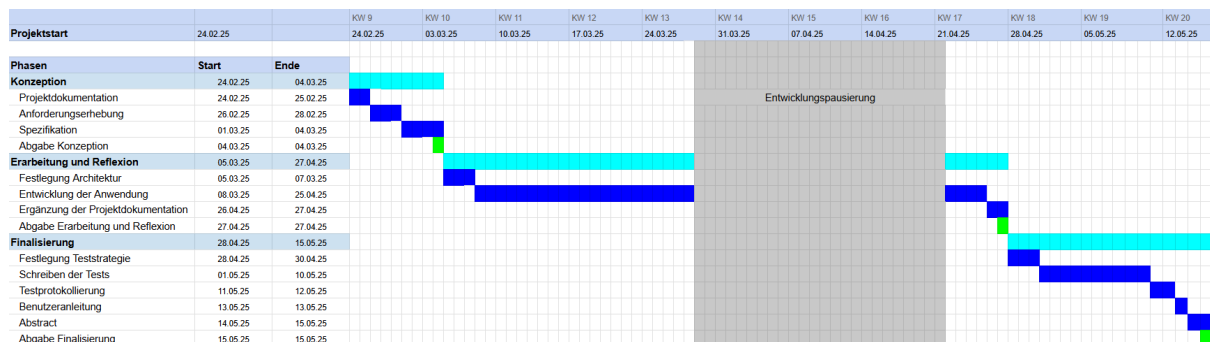
Schadensausmaß: Mittel

3. Link zum Github-Repository

https://github.com/djahan-latan/algorithm_visualizer

4. Zeitplan

alter Zeitplan zu Abgabe Phase 1:



aktualisierter Zeitplan zu Abgabe Phase 2:

