

Titel der Arbeit

Untertitel oder Titel der Arbeit in englischer Sprache

Masterarbeit

zur Erlangung des akademischen Grades Master of Science (M.Sc.)

Vorgelegt an der
Hochschule München
Fakultät für Informatik und Mathematik

Eingereicht von
Max Mustermann
Studiengang: Informatik
Matrikelnummer: 012345678

Erstprüfer: Prof. Dr. Markus Mustermann
Zweitprüfer: Maria Mustermann
Betreuer: Martin Mustermann
Abgabedatum: 01.01.2025

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne vertrauliche Informationen der Firma XYZ. Die Weitergabe des Inhalts der Arbeit im Gesamten oder in Teilen sowie das Anfertigen von Kopien und Abschriften sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Zustimmung der Firma.

Kurzfassung

Eine kurze Zusammenfassung des Zwecks, der Methoden, der wichtigsten Ergebnisse und der Schlussfolgerungen der Arbeit.

Danksagung

Ein optionaler Abschnitt, in dem Personen und Organisationen gedankt wird, die die Arbeit unterstützt oder dazu beigetragen haben.

Inhaltsverzeichnis

Abbildungsverzeichnis	VI
Tabellen-, Algorithmen- und Codeverzeichnis	VII
Abkürzungsverzeichnis	VIII
1 Einleitung	1
1.1 Abschnitt	1
1.1.1 Unterabschnitt	1
1.2 Forschungsfragen	1
1.3 Kapitelverweise	1
2 Verwandte Arbeiten	2
2.1 Tabellen	2
2.2 Zitate	2
3 Methodik	3
3.1 Abkürzungen	3
3.2 Abbildungen	3
4 Implementierung	4
4.1 Formeln	4
4.2 Algorithmen	4
4.3 Listen und Aufzählungen	4
4.3.1 Codebeispiele	5
5 Ergebnisse	6
5.1 Diagramme	6
6 Fazit	7
6.1 Gantt-Diagramm	7
Literatur	8
Ehrenwörtliche Erklärung	9
A Anhang	10
A.1 Zusätzliches Material	10

Abbildungsverzeichnis

3.1	DSR-Ansatz mit Relevanz-, Rigorositäts- und Design-Zyklus.	3
5.1	Ausführungszeit nach Größe der Trace-Datei des Frameworks.	6
6.1	Gantt-Diagramm des Projektzeitplans.	7

Tabellenverzeichnis

2.1	Ein- und Ausschlusskriterien für das Literatur-Review.	2
-----	--	---

List of Algorithms

4.1	Blockbasierte Dateiverarbeitung mit Hashing.	4
-----	--	---

List of Code Listings

4.1	Quellcode des Dateiverarbeitungsalgorithmus.	5
-----	--	---

Abkürzungsverzeichnis

DSR	Design Science Research	3
-----	-----------------------------------	---

1 Einleitung

1.1 Abschnitt

Dies ist ein Beispiel für einen Abschnitt.

1.1.1 Unterabschnitt

Dies ist ein Beispiel für einen Unterabschnitt.

Unterunterabschnitt

Dies ist ein Beispiel für einen Unterunterabschnitt.

1.2 Forschungsfragen

Dies ist ein Beispiel, wie man Forschungsfragen formatiert.

F₁: Ist dies eine Forschungsfrage?

Man kann auf Forschungsfragen wie folgt verweisen: F₁ ist ein Beispiel für eine Forschungsfrage.

1.3 Kapitelverweise

Man kann auf andere Kapitel wie folgt verweisen: Kapitel 6 ist das Abschlusskapitel.

2 Verwandte Arbeiten

2.1 Tabellen

Tabelle 2.1: Ein- und Ausschlusskriterien für das Literatur-Review.

Einschlusskriterien	Ausschlusskriterien
Englische Sprache	Nicht-englische Sprache
Veröffentlicht 2015–2025	Veröffentlicht vor 2015
Peer-reviewed	Nicht peer-reviewed
Liefert Methode und empirische Ergebnisse	Irrelevantes Fachgebiet oder Off-Topic

Man kann Tabellen erstellen und sie wie folgt referenzieren: Tabelle 2.1 zeigt die Ein- und Ausschlusskriterien für das Literatur-Review.

2.2 Zitate

Quellen können in Klammern wie folgt zitiert werden: Designmuster erleichtern die Wiederverwendung erfolgreicher Entwürfe und Architekturen [1].

Quellen können auch direkt im Text zitiert werden: Gamma u. a. [1] bietet einen umfassenden Überblick über Designmuster.

3 Methodik

3.1 Abkürzungen

Man kann Abkürzungen definieren und wie folgt darauf verweisen: Die Design Science Research (DSR)-Methodik wird häufig in der Forschung zu Informationssystemen verwendet. Der DSR-Ansatz betont die Erstellung und Evaluierung von Artefakten zur Lösung identifizierter Probleme.

3.2 Abbildungen

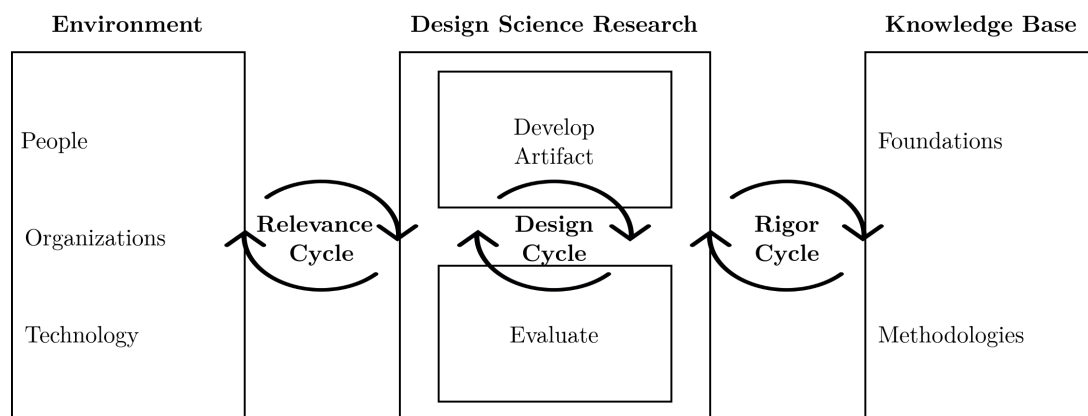


Abbildung 3.1: DSR-Ansatz mit Relevanz-, Rigorositäts- und Design-Zyklus. Adaptiert aus Hevner u. a. [2].

Man kann Abbildungen einfügen und wie folgt darauf verweisen: Abbildung 3.1 zeigt den DSR-Ansatz mit Relevanz-, Rigorositäts- und Design-Zyklus.

4 Implementierung

4.1 Formeln

Man kann Formeln wie folgt einfügen: Das Dateiverarbeitungsproblem kann wie folgt modelliert werden. Eine Datei F der Größe S Bytes, aufgeteilt in Blöcke mit maximaler Größe c , ist:

$$F = \langle B_1, B_2, \dots, B_m \rangle, \quad m = \left\lceil \frac{S}{c} \right\rceil$$

4.2 Algorithmen

Man kann Algorithmen mithilfe von Pseudocode wie folgt darstellen:

Algorithm 4.1: Blockbasierte Dateiverarbeitung mit Hashing.

Input: Datei F , Blockgröße c

Output: Datensatz $\langle size, block_count, sha256 \rangle$

```
1:  $h \leftarrow \text{INITSHA256}()$  ▷ Initialisiere Digest.
2:  $size \leftarrow \text{LENGTH}(F)$ 
3:  $block\_count \leftarrow 0$ 

4: for all  $block \in \text{READ}(F, c)$  do
5:    $\text{UPDATE}(h, block)$ 
6:    $block\_count \leftarrow block\_count + 1$ 

7:  $sha256 \leftarrow \text{FINALIZE}(h)$ 
8: return  $\langle size, block\_count, sha256 \rangle$ 
```

Man kann auf Algorithmen wie folgt verweisen: Algorithmus 4.1 beschreibt einen blockbasierten Dateiverarbeitungsalgorithmus.

4.3 Listen und Aufzählungen

Man kann Listen und Aufzählungen wie folgt erstellen:

1. Initialisiere einen SHA-256-Digest und lese die Dateigröße.

2. Lese die Datei in Blöcken von maximal c Bytes.
3. Aktualisiere für jeden Block den Digest und erhöhe den Blockzähler.
4. Finalisiere den Digest, um den Hash zu erhalten.
5. Gib $\langle size, block_count, sha256 \rangle$ zurück.

4.3.1 Codebeispiele

Man kann Quellcode einfügen, zum Beispiel Python-Code, wie folgt:

Listing 4.1: Quellcode des Dateiverarbeitungsalgorithmus.

```
1 import hashlib
2 from pathlib import Path
3
4 def process_file(path: str, chunk_size: int = 1 << 20):
5     p = Path(path)
6     h = hashlib.sha256()
7     with p.open("rb") as f:
8         while True:
9             b = f.read(chunk_size)
10            if not b:
11                break
12            h.update(b)
13     return {
14         "file": p.name,
15         "size_bytes": p.stat().st_size,
16         "sha256": h.hexdigest()
17     }
18
19 if __name__ == "__main__":
20     print(process_file("example.dat"))
```

5 Ergebnisse

5.1 Diagramme

Dies ist ein Diagramm, das mit Python (Jupyter Notebook) erstellt wurde, um den Stil des Dokuments beizubehalten.



Abbildung 5.1: Ausführungszeit nach Größe der Trace-Datei des Frameworks.

6 Fazit

6.1 Gantt-Diagramm

Dies ist ein Gantt-Diagramm, das mit draw.io erstellt wurde, um den Stil des Dokuments beizubehalten.

Activity	March	April	May	June	July	August
Define Research Scope						
Conduct Literature Review						
Implement Solution						
Evaluation and Refinement						
Thesis Writing						
Buffer						

Abbildung 6.1: Gantt-Diagramm des Projektzeitplans.

Literatur

- [1] E. Gamma, R. Helm, R. Johnson und J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1. Aufl. Addison-Wesley Professional, 1994, ISBN: 0201633612. Adresse: http://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612/ref=ntt_at_ep_dpi_1.
- [2] A. Hevner, A. R, S. March u. a., „Design Science in Information Systems Research,“ *Management Information Systems Quarterly*, Jg. 28, S. 75–, März 2004.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

München, 01.01.2025

Ort, Datum

Max Mustermann

Unterschrift

A Anhang

A.1 Zusätzliches Material