# Beschreibung sicherheitsrelevanten Verhaltens in der Open Source Softwareentwicklung

Präsentation Masterarbeit

Tobias Opel
Institut für Informatik
FU Berlin
02.10.2008

# Agenda

Freie Universität Berlin

1. Einleitung
   - Ziele und Aufgaben
   - Begriffe
2. Forschungsmethode Grounded Theory
3. Vorgehen
   - Datenerhebung
   - Offenes Kodieren
   - Phänomenbestimmung
   - Axiales Kodieren
4. Ergebnisse
   - Kernaussagen zur Phänomenuntersuchung
   - Vermeidung
5. Fazit

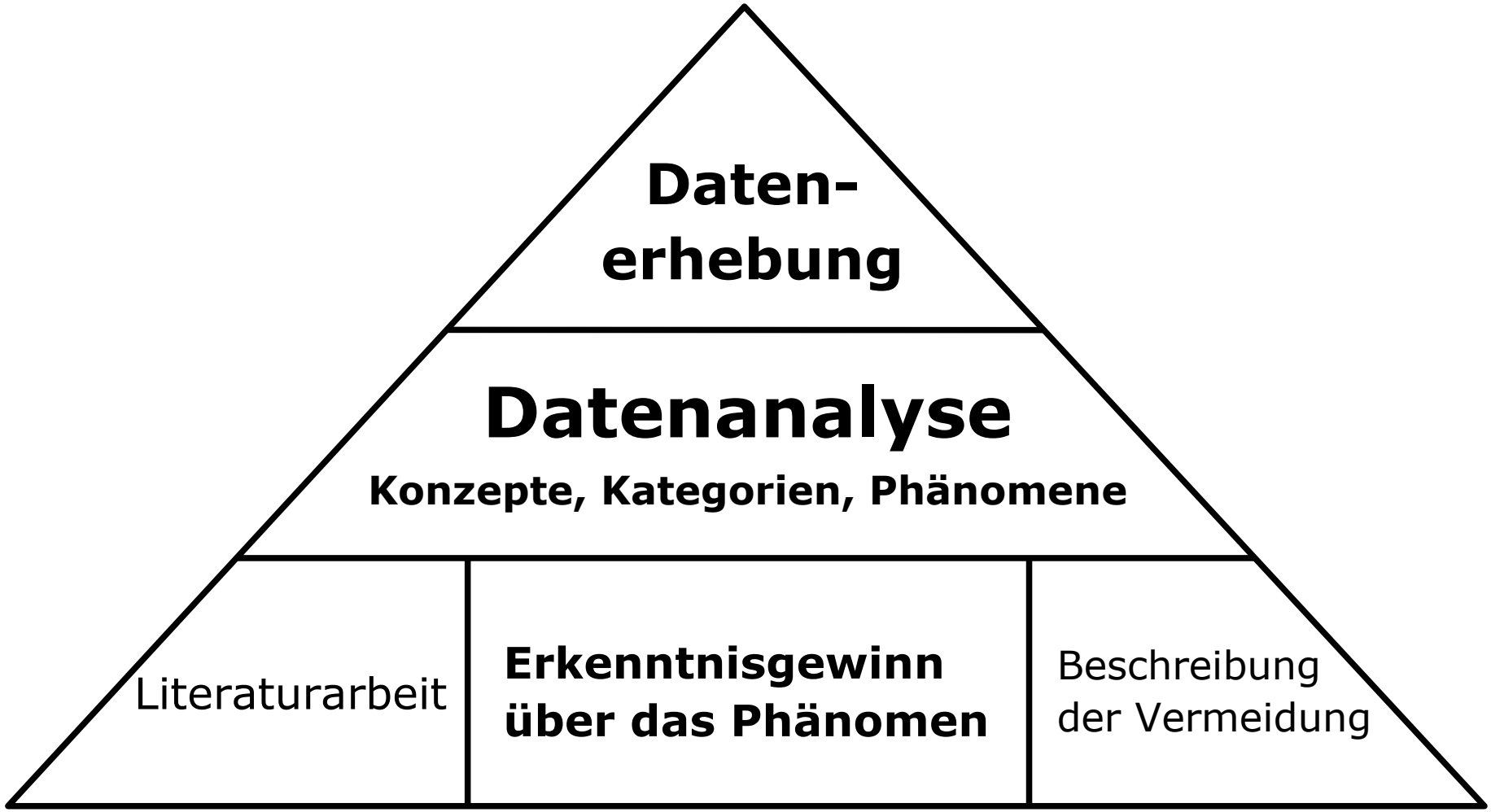
# Ziel der Arbeit

Freie Universität Berlin

- Beschreibung von sicherheitsrelevantem Verhalten in der Entwicklung von Open Source Software entwickeln
  - qualitative Datenanalyse existierender Datenbestände
  - Betrachtung der Behebung von sicherheitsrelevanten Schwachstellen
  - Identifikation und Klassifizierung von involvierten Verhaltensweisen und Rollen
  - Identifikation und Untersuchung dabei entdeckter Phänomene
  - Aussagen über die Erkenntnisse entwickeln

  - Beschränkung auf Open Source Webanwendungen

# Aufgaben


Daten-
erhebung
Datenanalyse
Konzepte, Kategorien, Phänomene
Literaturarbeit
Erkenntnisgewinn
über das Phänomen
Beschreibung
der Vermeidung

# Begriffe



- sicherheitsrelevante Schwachstelle
  - Defekt in der Software, welcher Sicherheit der Software einschränkt
  - Angreifer nutzt eine sicherheitsrelevante Schwachstelle aus
- sicherheitsrelevantes Verhalten
  - **Behebung** und Vermeidung einer sicherheitsrelevanten Schwachstelle
  - Teil der Qualitätssicherung innerhalb der Softwaretechnik
- Open Source Software unterliegt u.a:
  - uneingeschränkter Verbreitung
  - freier Verfügbarkeit des Quellcodes
  - erlaubter Modifikation und Ableitung des Quellcodes

# Forschungsmethode



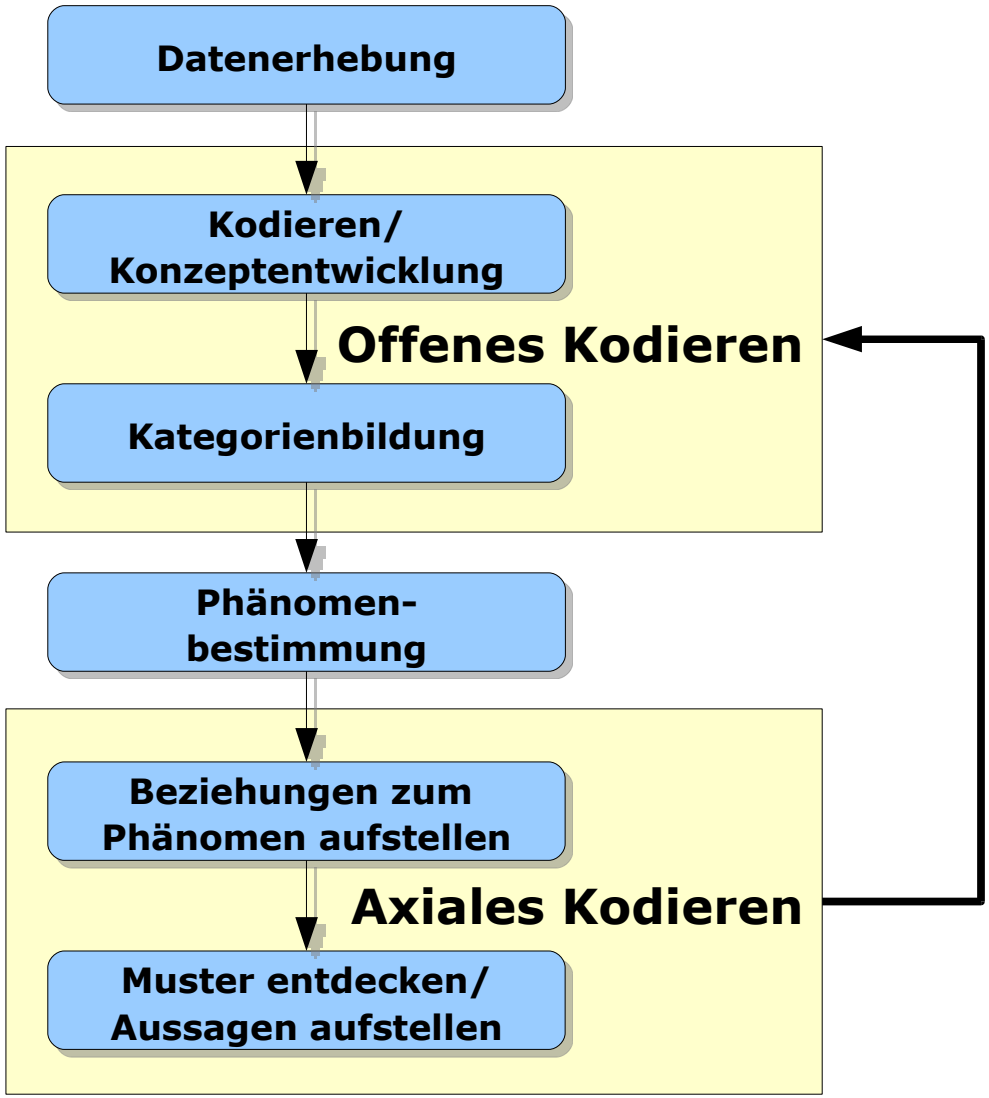
- Grounded Theory (gegenstandsverankernde Theoriebildung nach Strauss & Corbin [1]):
  - Qualitativer Forschungsansatz
  - Datenerhebung basierend auf dem zu untersuchenden Phänomen
    - Phänomen - Ereignis, Geschehnis, auf das eine Reihe von Handlungen oder Interaktionen gerichtet ist
- Anleitung zu drei Verfahren für methodisches, wissenschaftliches und schrittweises Vorgehen
- Offenes Kodieren
  - Konzeptualisieren der Daten (Kodieren)
  - Gruppieren der Konzepte zu Kategorien
  - Eigenschaften der Kategorien entwickeln
  - interessante Phänomene identifizieren

# Forschungsmethode



- Axiales Kodieren
  - Kategorien zum Phänomen in Beziehung setzen
    - Ursache – Phänomen – Konsequenz durch Strategien
    - diese Kategorien werden durch Kontext und Bedingungen beeinflusst
  - Verifikation dieser Beziehungen anhand der Daten und Suche nach weiteren Eigenschaften zu den Kategorien
  - Untersuchung der in Beziehung gesetzten Kategorien
    - konkrete Ausprägung der Eigenschaften dieser Kategorien in den Daten betrachten
    - ähnlich ausgeprägte Eigenschaften gruppieren und nach Einflüssen dafür untersuchen
- Selektives Kodieren (Nicht Teil meiner Arbeit)
  - dient der Theoriebildung

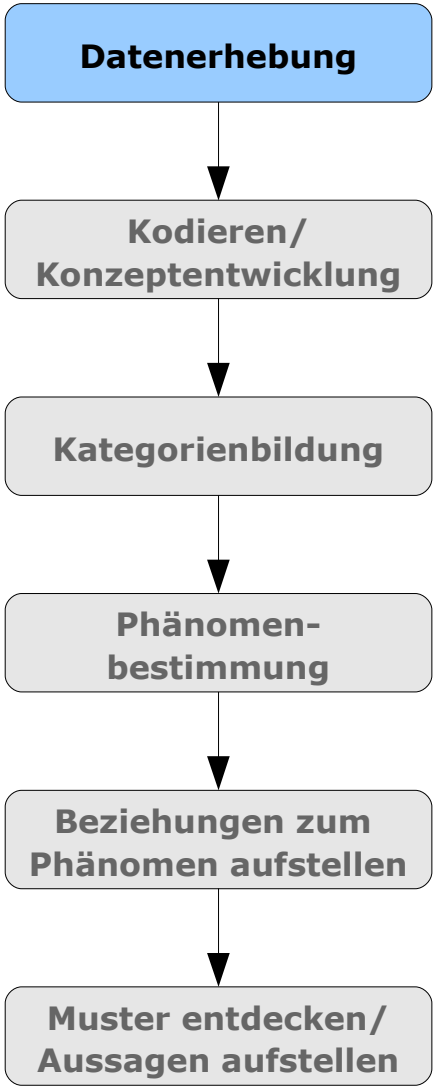
# Vorgehen


Datenerhebung
Kodieren/
Konzeptentwicklung
Offenes Kodieren
Kategorienbildung
Phänomen-
bestimmung
Beziehungen zum
Phänomen aufstellen
Axiales Kodieren
Muster entdecken/
Aussagen aufstellen

# Datenerhebung - Projektauswahl

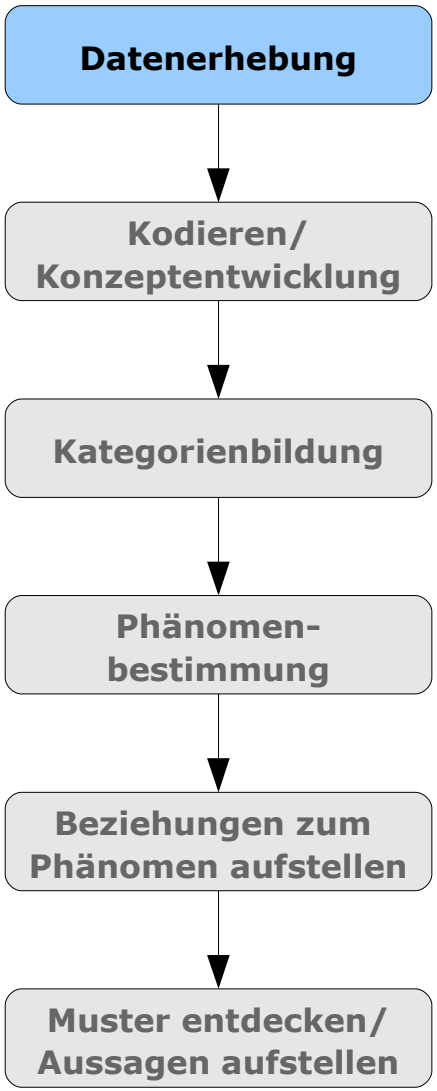
Freie Universität Berlin

- Datenbanken für Schwachstellen (z.B. NVD)
  - Filterung der Ergebnisliste nach:
    - Zeitraum (01.01.2007 - 01.03.2008)
    - Open Source Software
    - Webanwendungen
    - Behobene Schwachstellen
- Newsseiten zur IT-Sicherheit
- Projekte aufgrund eigener Erfahrung
- erste mögliche Kandidaten:
  - Drupal
  - Horde
  - Wordpress
  - Joomla



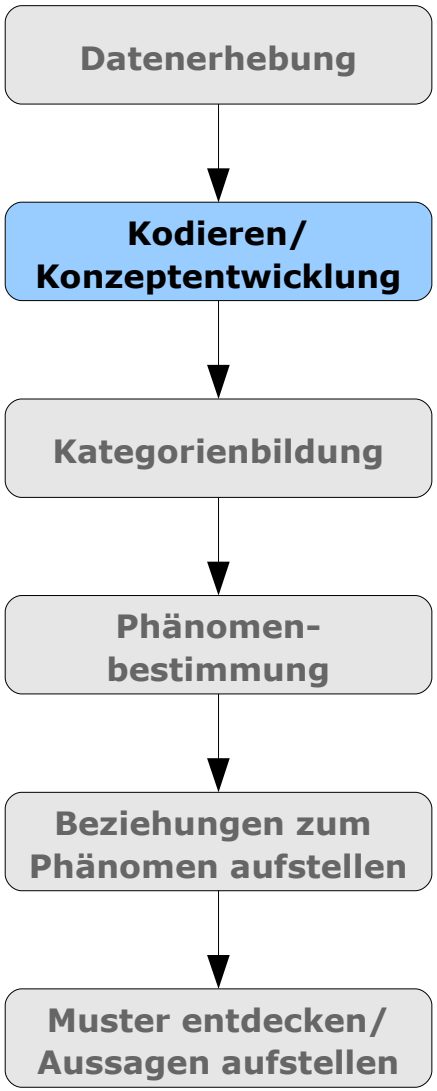
# Datenerhebung - Datenbasis

- Daten zu Projekten aus verschiedenen Quellen gesammelt
  - z.B. Mailinglisten, Foren, Blogs, Tracker
- Daten zu einer Schwachstelle und deren Behebungsverlauf zu einer **Episode** zusammengefasst
- Projekte zu Beginn der Analyse:
  - ~~Drupal~~ (Zu wenig Daten Vorhanden)
  - ~~Horde~~ (Zu wenig Daten Vorhanden)
  - Wordpress (5 Episoden)
  - Joomla (9 Episoden)



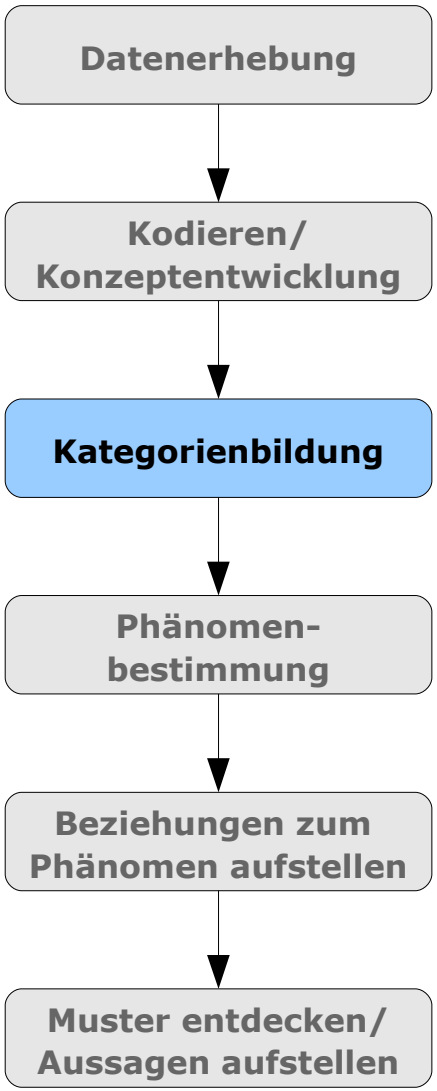
# Offenes Kodieren

- Verhaltenskonzepte
  - offenes Kodieren von fünf Wordpress Episoden und einer Joomla Episode
  - dabei erste Phänomene entdeckt
  - 110 unterschiedliche Konzepte entwickelt
    - z.B. bewertet_Schweregrad, stelltBereit_ Exploit, gibtAn_Exploitfundort(e)
- Rollenkonzepte
  - aus Verhaltenskonzepten, Hinweisen im Text oder Metadaten zu dem Projekt
  - 14 unterschiedliche Konzepte entwickelt
    - z.B. Kernteammitglied, Schwachstellen-Berichterstatter, Hilfeleistender



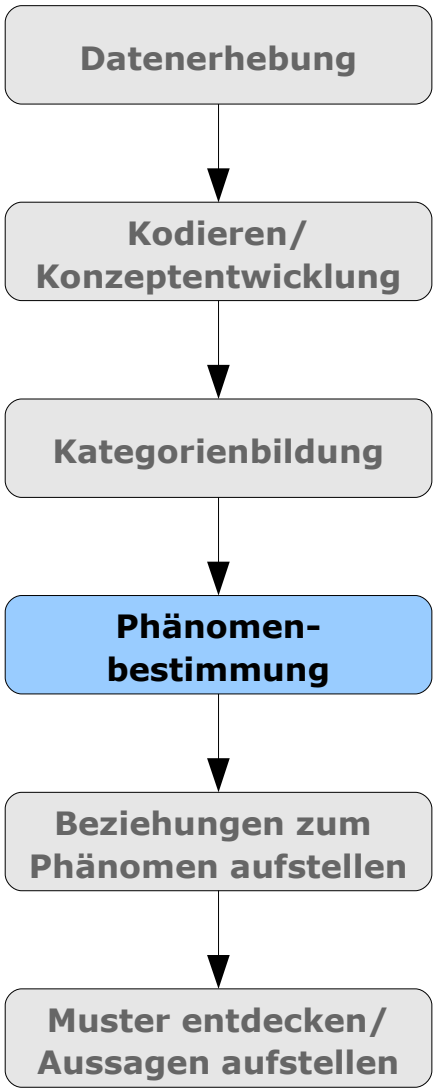
# Offenes Kodieren

- Verhaltenskategorien
  - Gruppierung von zu einem Phänomen gehörender Verhaltenskonzepte
  - 24 unterschiedliche Kategorien entwickelt
    - z.B. Schwachstellendarstellung, Umgang mit voller Schwachstellenveröffentlichung
- Rollenkategorien
  - bleiben für Akteure gleich unabhängig von den angenommenen Rollenkonzepten
  - 3 unterschiedliche Kategorien entwickelt
    - betroffene/besorgte Nutzer, externe Helfer, Verantwortliche des Herstellers



# Phänomenbestimmung

- Untersuchung des Phänomens "Umgang mit Schwachstellenveröffentlichung"
  - öffentliche Darstellung aller Details zur Schwachstelle durch externen Akteur
    - Schwachstellenbeschreibung, betroffene Version, Proof of Concept, Ausnutzung, Folgen, Sicherheitsrisiko, Eingrenzung, (Ab-)Sicherung
- Warum dieses Phänomen?
  - tritt in vielen Episoden auf
  - als eigene Verhaltenskategorie entwickelt
  - in der Literatur diskutiert
    - richtige Vorgehensweise bei Veröffentlichung
    - Untersuchung der Auswirkung

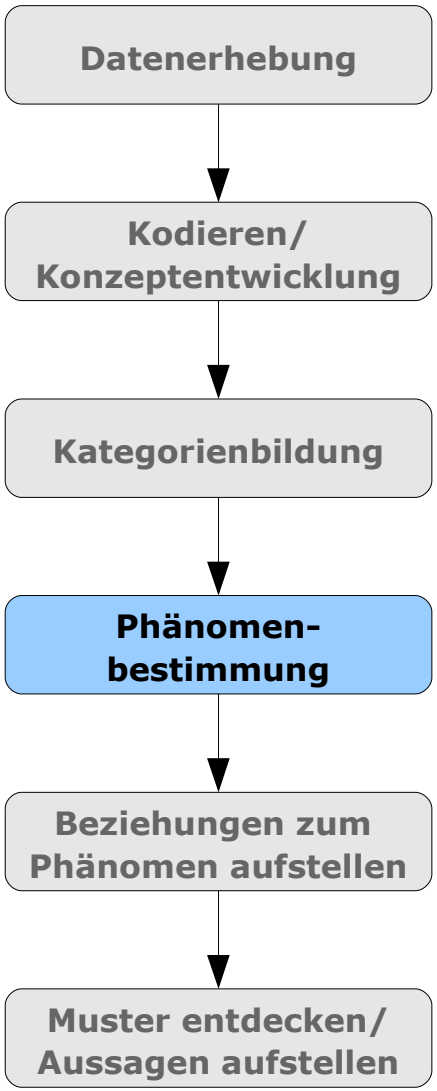


# Phänomenbestimmung

- Ziele der Phänomenuntersuchung

  *Gemeinsamkeiten in meinen Daten zu finden, welche den Einfluss auf die Behebungsdauer, den Behebungsverlauf und den Einfluss der verschiedenen Rollen beschreiben.*

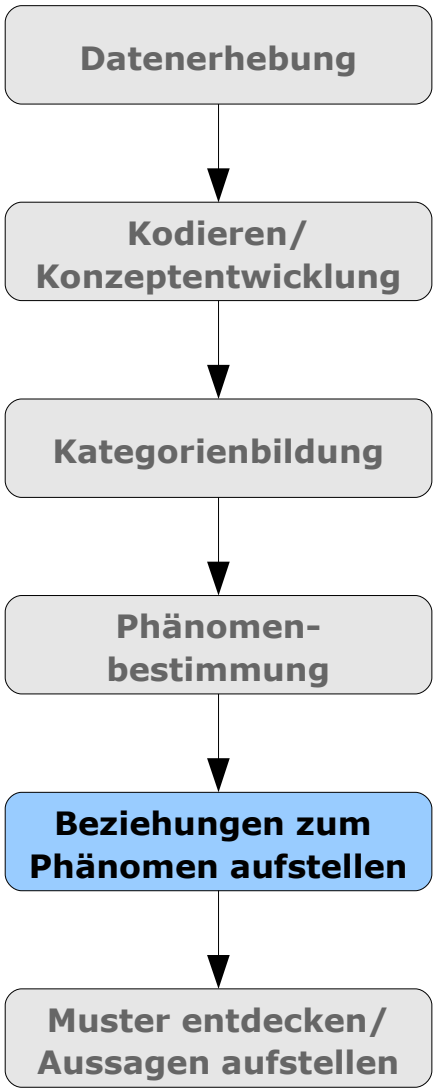
- Behebungsdauer nach Witten et al. [2]
  - einfache Methode zur Bewertung der Sicherheit
  - Zeitraums zwischen der Veröffentlichung einer Schwachstelle und deren Behebung
  - Dauer als Vergleichskriterium
  - kurze Dauer ist besser



# Axiales Kodieren

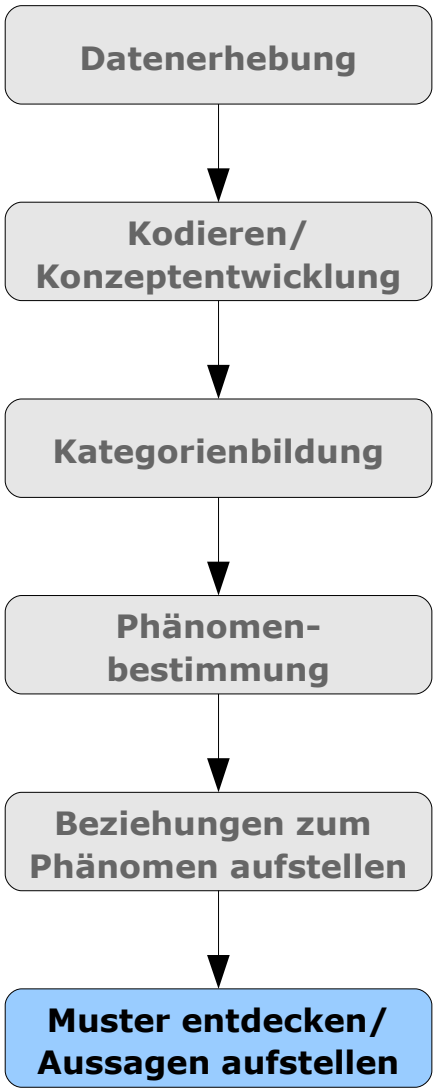
- Ursache *Schwachstellenveröffentlichung*
- Phänomen *Umgang mit der Schwachstelle nach Veröffentlichung*.
- Konsequenz *Schwachstellenbehebung*
- Kontext unter dem Aspekt der verschiedenen *Rollen* und *Eigenschaften*
  - z.B. Behebungsdauer
- weitere Bedingungen wie z.B. *Konflikte*

- Untersuchung einzelner Eigenschaften in den Daten nach Gemeinsamkeiten
- Gemeinsamkeiten führen zu Aussagen über Beziehungen zwischen Eigenschaften



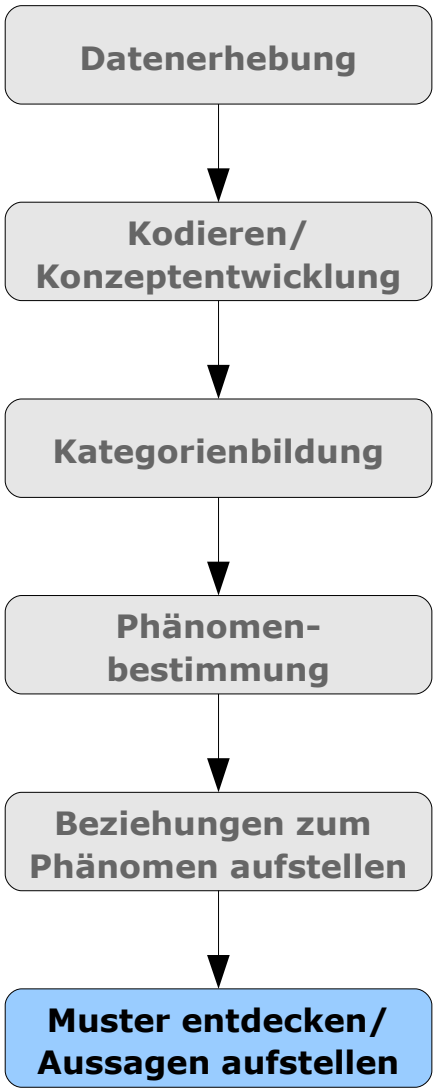
# Axiales Kodieren

- Ursache *Schwachstellenveröffentlichung*
  - Betrachtung der Eigenschaften
    - Angabe der Ausnutzung, (Ab-)Sicherung, Folgen, Eingrenzung
  - Schwachstellenbehebung in 13 von 16 Episoden innerhalb eines Monats
  - in 12 der 13 Episoden wurde eine Ausnutzung der Schwachstelle angegeben
  - **Aber** auch in 2 von den 3 Episoden mit der Behebungsdauer über einem Monat



# Axiales Kodieren

- daher Erweiterung der Untersuchung auf den Kontext des Phänomens
- Betrachtung der Eigenschaft "*Subjektive Einstufung des Sicherheitsrisikos*"
  - das Schwachstellenrisiko in beiden Episoden wurde vom Hersteller als niedrig eingestuft
  - ebenso bei der dritten Episode mit Behebungsdauer über einem Monat
- beide Erkenntnisse führen zu den Aussagen 1 und 2 auf der folgenden Folie



# Ergebnisse



- Kernaussagen zur Phänomenuntersuchung
  1. Hersteller reagierte auf schwerwiegend eingestufte Schwachstelle schneller als auf niedrig Eingestufte
  2. Angabe einer Schwachstellenausnutzung führte zu einer schnellen Behebung (sofern nicht durch 1. eingeschränkt)
  3. schnellere Behebung bei Projekten mit direktem Veröffentlichungskanal und expliziter Kennzeichnung der Schwachstelle
  4. Veröffentlichung der Behebung gekoppelt an einen Release dauerte länger als die Veröffentlichungen als Zwischenpatch
  5. externe Helfer geben Nutzern Informationen und Schutz-maßnahmen und leisten Überzeugungsarbeit beim Hersteller

# Ergebnisse



- Verhaltensweisen und Praktiken zur Vermeidung
  - Quellcodereview nach ähnlichen Schwachstellen aufgrund eines Schwachstellenfunds
  - Übernahme von Praktiken zur Schwachstellenvermeidung aus anderen Open Source Anwendungen
  - Bereitstellung von Anleitungen und Durchführung von sicherem Konfigurieren der Anwendung
  - Anleitung für sicheren Code, um Auftreten bestimmter Schwachstellenarten zu vermeiden

# Fazit



- Fazit und Erkenntnisse
  - erstes Mal wissenschaftlich qualitative Forschung betrieben
  - Entwicklung einer Beschreibung von sicherheitsrelevantem Verhalten in der Open Source Softwareentwicklung
    - Verhaltens- und Rollenklassifikation erstellt und dabei Phänomene identifizert
    - Klassifikation für Beziehungen zum Phänomen herangezogen
    - durch Untersuchung zum Phänomen Erkenntnisse zu Behebungsdauer, Behebungsverlauf und Rollen gewonnen
  - viel über Sicherheit und Open Source Software gelernt

# Fazit



- Ausblick und weiteres Vorgehen
  - weitere Phänomene untersuchen
  - Projekte anderer Art von Software (z.B. Bibliotheken, Betriebssysteme) untersuchen
  - andere Datenquellen einbeziehen (z.B. Ergebnisse von Interviews, Umfragen oder Fallstudien)
  - Erkenntnisse und Untersuchungen auf Closed Source Software erweitern

# Literatur



[1] Anselm Strauss und Juliet Corbin. *Grounded Theory: Grundlagen Qualitativer Sozialforschung*. Beltz Psychologie Verlags Union, 1996.

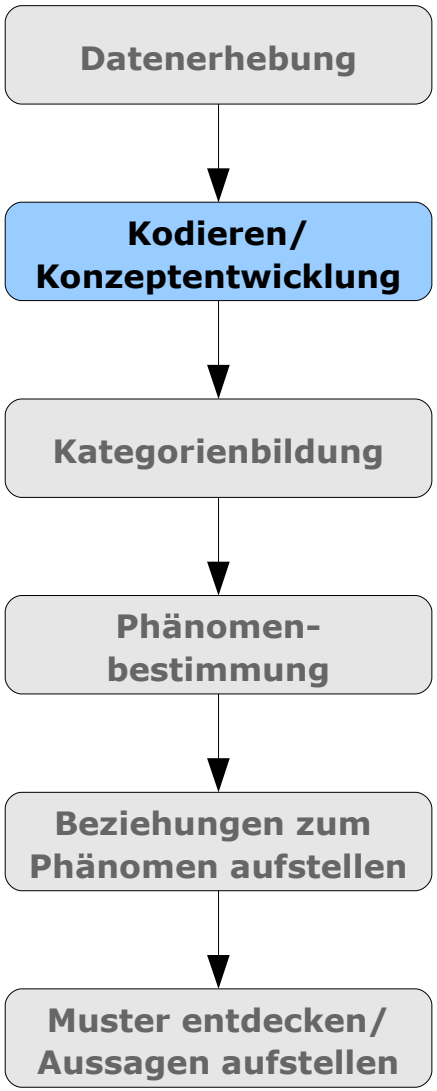
[2] Brian Witten, Carl Landwehr, und Michael Caloyannides. *Does open source improve system security?* IEEE Software, 18(5):57–61, 2001. ISSN 0740-7459.

# Vielen Dank!

# Verhaltensklassifikation



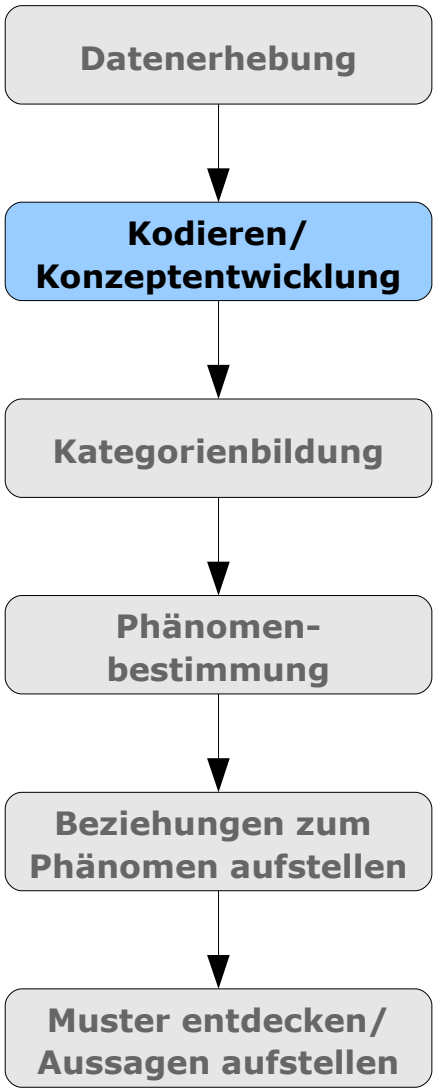
- Erstellung einer Klassifikation für Verhaltensweisen während der Behebung
  - Hierarchische Anordnung nach Konzepten, Kategorien, Hauptkategorien
  - Blick auf Verhaltensweisen, welche innerhalb des Behebungsverlaufs auftreten können
- Verhaltenskonzepte
  - entstanden durch offenes Kodieren von fünf Wordpress Episoden und einer Joomla Episode
  - dabei erste Phänomene entdeckt
  - 110 unterschiedliche Konzepte entwickelt
    - z.B. bewertet_Schweregrad, stelltBereit_ Exploit, gibtAn_Exploitfundort(e)



# Rollenklassifikation



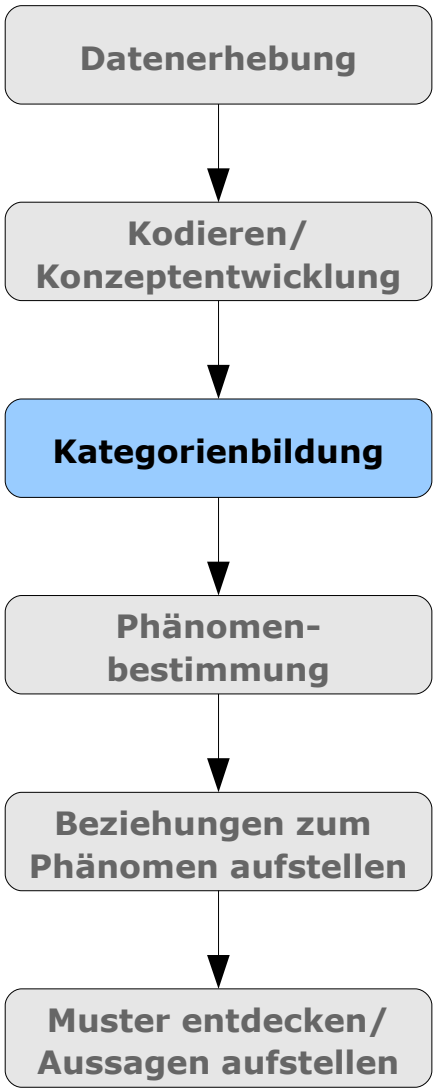
- Rollenkonzepte
  - Verhaltenskonzepte eines Akteurs
  - Hinweise im Text
  - aus Metadaten zu dem Projekt
  - 14 unterschiedliche Konzepte entwickelt
    - z.B. Kernteammitglied, Schwachstellen-Berichterstatter, Hilfeleistender



# Verhaltensklassifikation

Freie Universität Berlin

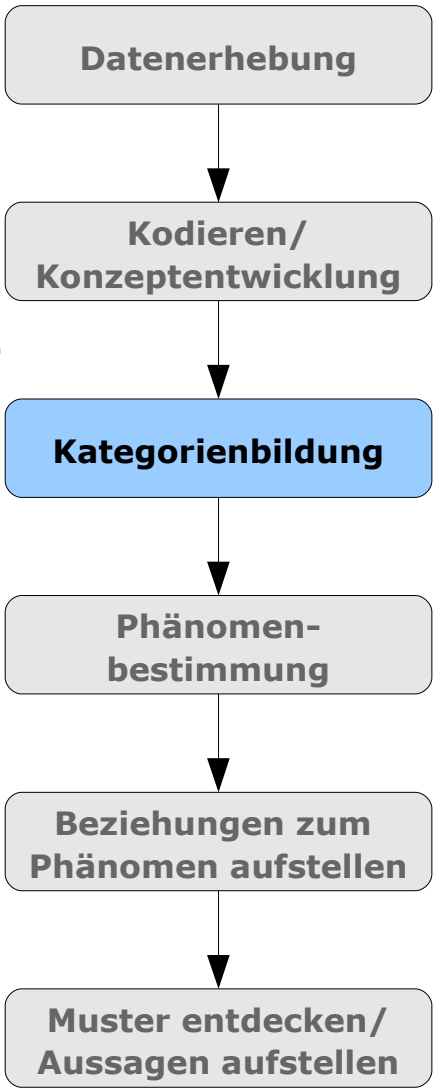
- Verhaltenskategorien
  - Gruppierung zu einem Phänomen gehörende Verhaltenskonzepte
    - aufgrund gleicher Aktivitäten innerhalb der Konzepte
    - aufgrund gleicher Objekte, auf denen die Aktivitäten gerichtet sind
    - aufgrund des Kontextes bzw. Auftreten im Behebungsverlauf
  - 24 unterschiedliche Kategorien entwickelt
    - z.B. Schwachstellendarstellung, Umgang mit voller Schwachstellenveröffentlichung,



# Verhaltensklassifikation

- Hauptkategorien
  - liefern speziellen Blick auf die Daten und enthaltene Phänomene
  - weitere Abgrenzung der Verhaltenskategorien
  - bieten bessere Übersicht für Einordnung neuer Konzepte
  - Gruppierung für weitere Untersuchungen

  - **Informationsbereitstellung**
  - **Informationsverwertung/-verarbeitung**
  - **Einleitung konkreter Maßnahmen**
  - **Umgang mit Konflikten**
  - **Unterstützung**
  - **Einsatz von Werkzeugen**

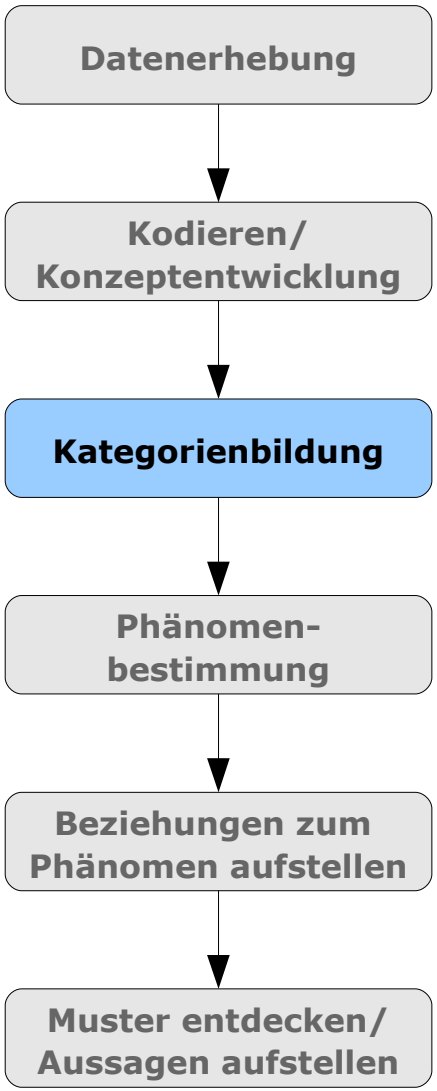


# Rollenklassifikation



- Rollenkategorien
  - Akteur kann während des Behebungsverlaufs verschiedene Rollen(-konzepte) annehmen
  - Kategorien bleiben für Akteure gleich unabhängig von den angenommenen Rollenkonzepten

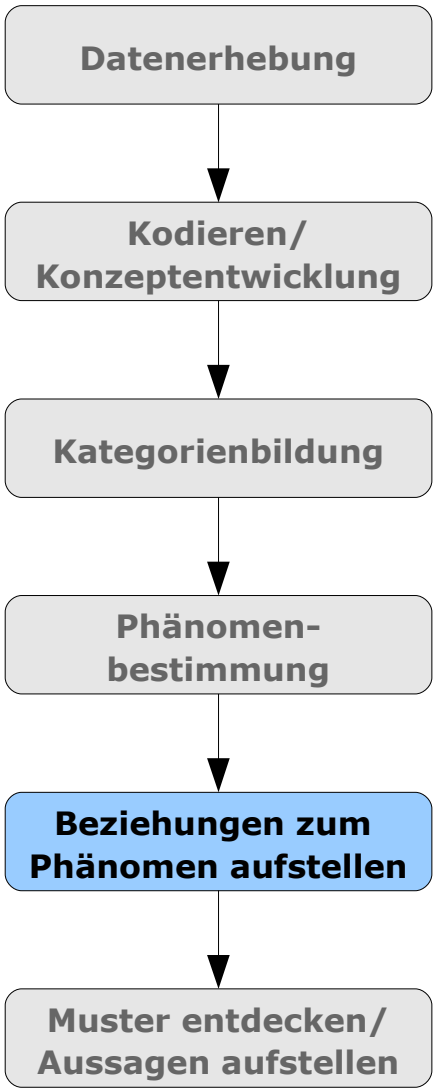
  - **der betroffene/besorgte Nutzer**
  - **der externe Helfer**
  - **der Verantwortliche des Herstellers**



# Phänomenuntersuchung



- Die (**Ursächliche Bedingung**) *Schwachstellenveröffentlichung* führt zum
- (**Phänomen**) *Umgang mit der Schwach-stelle nach Veröffentlichung*.
- Der (**Kontext**) stellt den
  - spezifischen Satz der Eigenschaften des Phänomens
  - unter dem Aspekt der *Rollenkategorien* dar.
- Die (**angewandten Strategien**) zur *Behebung* bzw. *Folgenminimierung*
  - beeinflusst durch (**intervenierende Bedingungen)** wie *Konflikte*
- führen zur (**Konsequenz**) *Behebung.*



- Kernaussagen und Ergebnisse I

1. Hersteller reagierte auf schwerwiegend eingestufte Schwachstelle schneller als auf niedrig Eingestufte
2. Angabe einer Schwachstellenausnutzung führte zu einer schnellen Behebung (sofern nicht durch 1. eingeschränkt).
3. Projekte mit direktem Veröffentlichungskanal und expliziter Schwachstellenkennzeichnung behoben schneller.
4. Öffentlich gemeldete Schwachstellen auf einem inoffiziellen Veröffentlichungskanal hatten eine kurze Behebungsdauer
5. fehlte die konkrete Angabe aller betroffenen Versionen, so kam es zu einer vergleichsweise längeren Behebungsdauer

- Kernaussagen und Ergebnisse II

6. Veröffentlichung der Behebung gekoppelt an einen Release dauerte länger als die Veröffentlichungen als Zwischenpatch
7. Einsatz qualitätssichernder Praktiken erhöhte die Behebungsdauer
8. Drängen in direkter oder unkonventioneller Form führte zur Reaktion des Hersteller
9. externe Helfer geben Nutzern Informationen und Schutz-maßnahmen und leisten Überzeugungsarbeit beim Hersteller
10. Keine Erkenntnisse über den Einfluss der Angabe von „(Ab-)Sicherungen“ bzw. von „Schwachstellenfolgen“

# Fazit



- Herausforderungen
  - zu Beginn kein konkretes Vorgehen bei der Datenanalyse
    - Forschungsmethode Grounded Theory ausgewählt
  - Datenerhebung durch Sicherheitsteams und interne, geschlossene Mailinglisten beeinträchtigt
    - andere Projekte gesucht
  - erstellte Klassifikation zu allgemein
    - Auswahl und Untersuchung konkreter Phänomene
  - zu wenig Zeit um mehr als ein Phänomen zu untersuchen
    - weitere Phänomene in das Untersuchte integriert
    - weitere Phänomene als Ausblick für weitere Untersuchungen dargestellt