

The Normative Fallacy in Software Engineering Research

Teil 2: Diskussion mitgebrachter Beispiele und Fazit

Seminar: Beiträge zum Software Engineering
2020-05-25

Erinnerung

- Sowohl **deskriptives** als auch **normatives** Denken ist wertvoll in empirischer Software Engineering-Forschung
- Problematisch wird es
 - wenn diese Unterscheidung nicht gemacht wird,
 - und **normative Theorie** für **deskriptiven Zweck** verwendet wird
- **Normative Fallacy:**
 - Die **Annahmen** in verwendeten Konzepten sind empirisch nicht validiert.
 - Diese prägen **Datenerhebung** und **-interpretation**
 - und daraus werden **irreführende** Schlussfolgerungen gezogen.

Erinnerung (2): Beispiele

- Entscheidungsfindung:
 - Rational decision-making (Faktoren, Gewichte) vs. Naturalistic decision-making (adaptiv, basierend auf Erfahrung und gedanklicher Simulation)
[Becker, 2020]
- Technische Schulden:
 - Kontext-freie statische Quellcode-Metriken
- Integrationstests:
 - Test-Pyramide (Unit-, Integrations-test, Ende-zu-Ende-Test)
- Pair Programming:
 - Zwei Entwickler mit klaren Rollen arbeiten wie ein Verstand (und für alle bedeutet das das Gleiche)

Offener Teil: Die mitgebrachten Beispiele

Beispiel 1

- What are your programming language's energy-delay implications?
 - Falsche Annahme: Programmierer können die Programmiersprache falsch auswählen
 - Die Studie ändert sich dadurch aber nicht
- Dies ist also eher kein Beispiel für eine Normative Fallacy

Beispiel 2

- [Predicting Usefulness of Code Review Comments Using Textual Features and Developer Experience](#)
 - Falsche Annahme: Code-Review-Kommentare, denen keine Codeänderung folgt, sind nutzlos
 - Unglaublicher Unfug
 - Dadurch wird die ganze Studie sinnlos, denn es geht die ganze Zeit nur diese Unterscheidung:
Kommentare mit/ohne anschließende Codeänderung

Beispiel 3

- [Why Does Code Review Work for Open Source Software Communities?](#)
 - Verdacht: Machen die Autor_innen die Annahme, dass Code Review nur Vorteile hat und übersehen mögliche Nachteile?
 - Die Studie sucht kaum nach Nachteilen, aber es ist nicht klar, ob das ein Problem darstellt.

Eine Art Fazit

- Die Fallacy vermeiden:
 - Forschende sollten sich ihrer Annahmen bewusst werden
 - "Den Daten zuhören"
 - Offene Fragen stellen
 - Erfahrenen Praktiker/innen aufmerksam zuhören
 - Qualitativer Forschungsansatz
- [Becker, 2020]
- Dennoch: Nutzen nicht-empirischer Konzepte
 - Verwendung in zu prüfender Hypothese
 - Im Gespräch mit Praktikern
 - um überhaupt etwas konkreteres fragen zu können als *"Und? Wie ist es so?"*
 - z.B. *"Also, 'Integrationstests' haben wir hier nicht, aber ... und ...!"*

Ergänzungen?

Offen: Wie nennen wir das Kind?

- Original "Normative Fallacy"
 - Campbell (1970): Kritik an Moralphilosophen, die ihre Analysen auf ihren eigenen **Überzeugungen** aufbauen und ihre Schlussfolgerungen wie **Fakten** präsentieren
 - Gemeint als Gegenstück zur "Naturalistic Fallacy"
 - dem Fehlschluss, dass sich Werte aus "natürlichen" Gegebenheiten ableiten lassen
- Verwandte Begriffe:
 - Glasers "*Forcing*" in qualitativer Forschung: Das Aufzwingen von vorgefassten Konzepten auf die Daten
 - Statt: Emergierende, "auftauchende" Konzepte aus den Daten selbst
 - Bestätigungsneigung, *confirmation bias*: Informationen so auswählen, dass diese die eigenen Erwartungen bestätigen

Nun?

Referenzen

- Becker (2020): [Beware the Normative Fallacy](#) (arXiv)
- Campbell (1970): [The Normative Fallacy](#) (Philosophical Quarterly)