Combining Qualitative & Computational Methods: Two Stories

Aron Lindberg
Figure 1. Abductive Inquiry as Discovery and Justification

Qualitative $\rightarrow$ Computational
Discursive Modulation in Open Source Software: How Communities Shape Novelty and Complexity

Forthcoming in MIS Quarterly

Lindberg, Berente, Howison, and Lyytinen
Discursive Modulation in Open Source Software

- We examine how two OSS projects engage in distinct “discursive modulation practices” to shape software novelty and complexity.
- Based on our research we formulate a theory that explains how modulation results in alternative OSS community approaches to shaping software novelty and complexity, and how this process reflects, and is reflected in, the resulting software artifact.
- The identification of modulation practices is done through qualitative analysis, while the software artifacts are examined using codebase analysis.
Discourse

produces specific combinations of
discursive positions

Filtering
Mixing

selects discursive positions

mutual shaping

Artifact

constrains new feature development

Novelty
Complexity

creates new interdependencies
Data

- Interviews
- Podcasts
- Industry conference presentations, workshops, panels, etc.
- Blogs
- Mailing list discussions
- Issues and pull requests on GitHub

Lesson: Different types of trace data offer different degrees of detail and exposition!
Method

- Qualitative analysis was used to show how a particular set of core doctrines and design principles were enacted in practice.
- Computational techniques were used to capture novelty and complexity (i.e., concept operationalization) and show how the structure of artifacts related to enactments.

**Lesson:** Computational crafting of concept operationalizations is hard work!
Framing

● Multiple refraamings:
  ○ From epistemic stances to discursive modulation
  ○ From generativity & technical debt to novelty & complexity
  ○ Entailed iterative revision of computational operationalizations of concepts

Lesson: While reframing a paper is risky, it can help the different parts of a manuscript align better with each other!
Theorizing

- Iterative across computational and qualitative evidence to achieve consistency
- Recrafting of computationally measured concepts went hand in hand with reframing of qualitative results

Lesson: The research process is often more iterative than it looks in the manuscript!
Review Process

- 2 rounds at ISR
- 3 rounds at MISQ
- Major reframing at both journals

**Lesson:** Rejections & harsh criticism does not necessarily predict doom & gloom for your manuscript!
Computational -> Qualitative
Forthcoming in MIS Quarterly

The Entrainment of Task Allocation and Release Cycles in Open Source Software Development

Lindberg, Schecter, Berente, Lyytinen, and Hennel
The Entrainment of Task Allocation and Release Cycles in Open Source Software Development

- We identify a process of “entrainment” around open source software (OSS) development release cycles to capture patterns of self-organized task allocation among developers.
- We conducted an abductive, computationally-intensive study of eight OSS projects, using relational event modeling to analyze 1,169,489 actions covering 93 major software releases.
- We use qualitative analysis to understand the mechanisms which generate entrainment of task allocation and release cycles.
- Our theorizing suggests that release cycles constitute important events around which OSS communities self-organize and we characterize how this occurs.
Figure 7. Oscillating task allocation across the release cycle

Notes. The increasing saturation of the coloring of each state indicates increasing developer-issue inertia
Data

- Digital trace data of development actions
- Interviews

Lesson: Multi-method/data studies often need large research teams with multiple competencies!
Method

- Computational analyses (i.e., relational event modeling) used to show evolution of task allocation over time
- Qualitative analysis was used to theorize generative mechanisms behind computationally identified patterns

**Lesson:** Qualitative method is useful for adding richness to thin computational analyses as well as providing the basis for theorizing generative mechanisms!
Framing

- Initial framing: developer attention
- During the review process we shifted to entrainment

**Lesson:** Trace data often benefits from focus on behavior as opposed to cognition or affect!
Theorizing

- Theorizing was based on intimate knowledge of qualitative data (from my doctoral thesis) which helped to interpret the relational event modeling results.
- The actual qualitative evidence was added in the third revision round.

**Lesson:** Qualitative data facilitates interpretation of statistical results!
Review Process

- It took a lot of time to get the manuscript from an AOM OCIS best conference paper award (2015) to submission to MISQ (2020)
- Adding co-authors with data scraping and computational analysis skills was crucial
- Review process was relatively straightforward but involved collecting additional trace data to improve robustness, simulations as validity checks, etc.
- Qualitative data was added in the last round of revisions

**Lesson:** Be prepared to collect additional data during the review process!
THANK YOU

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030