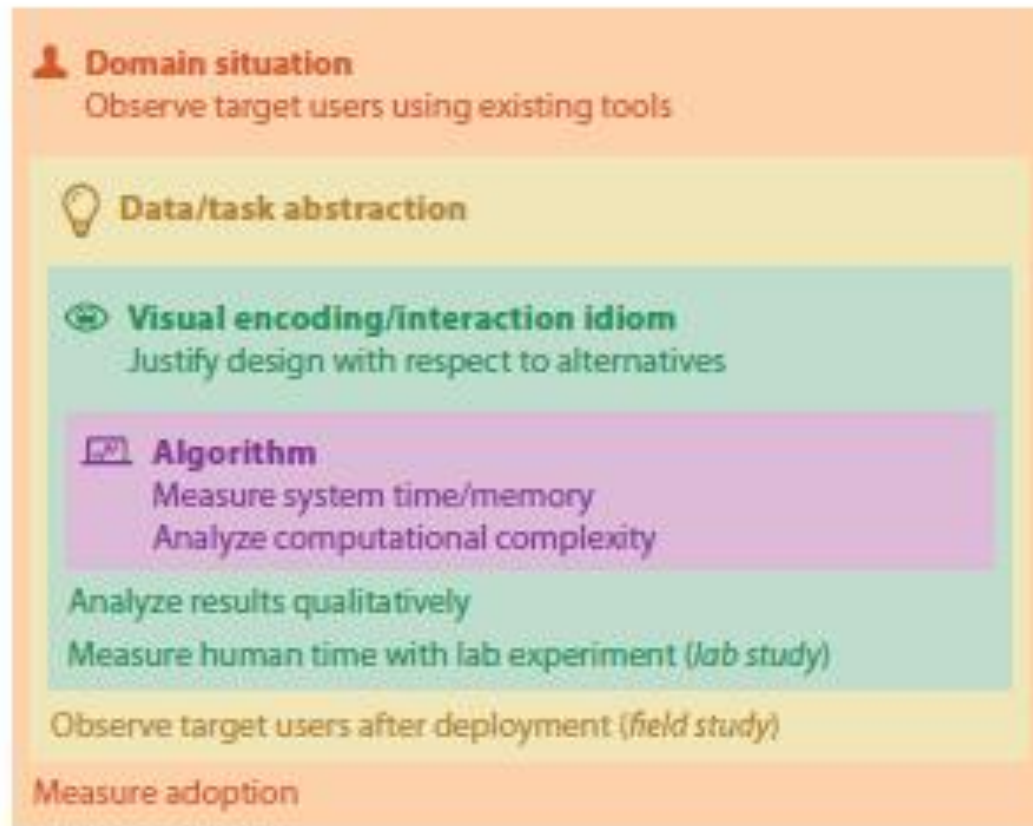# The Different threats of Validity

Figure 4.1. The four nested levels of vis design have different threats to validity at each level, and validation approaches should be chosen accordingly.
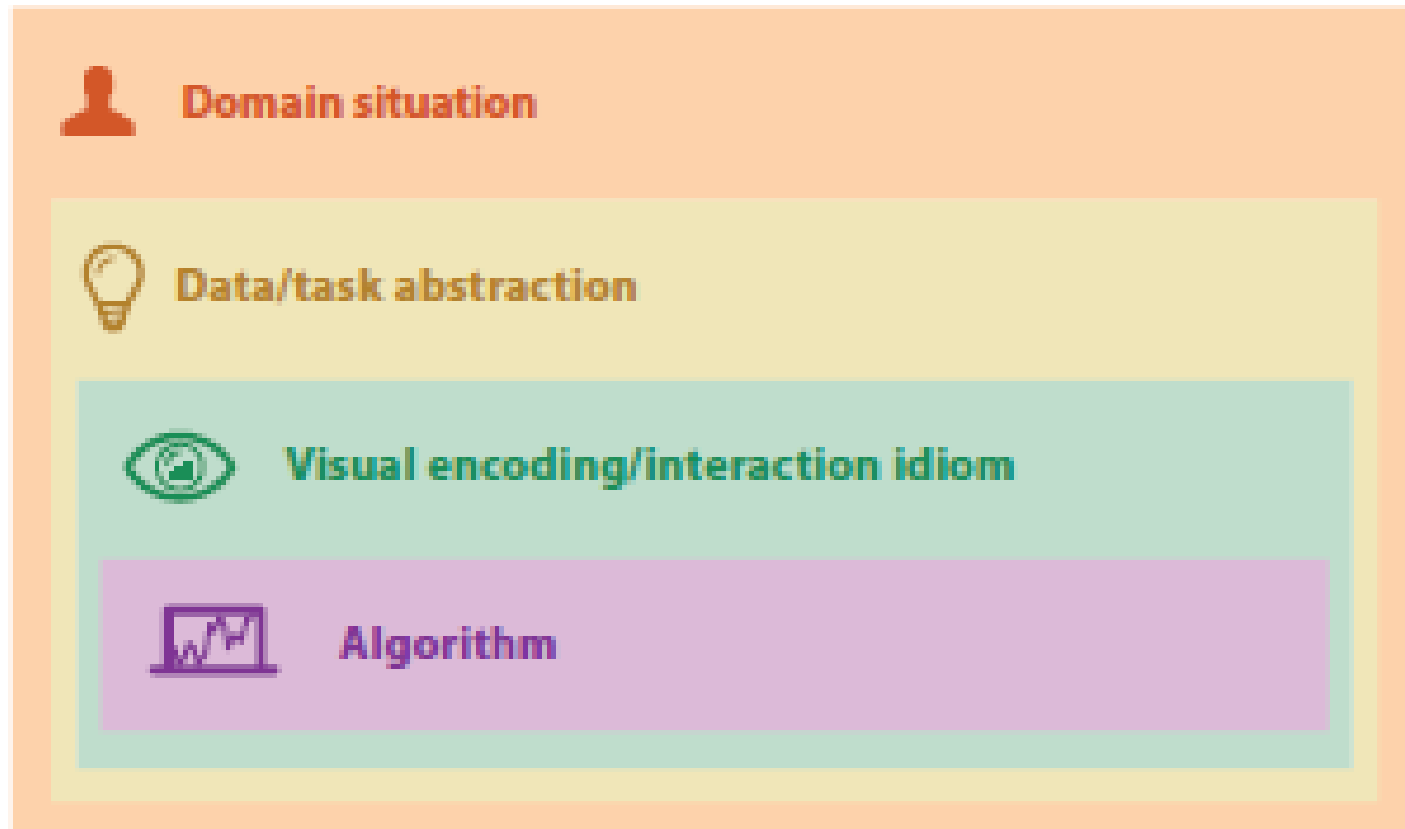
The four nested levels of vis.design

Figure 4.2. The four nested levels of vis design.

★ Working closely with a specific target audience to iteratively refine a design is called user-centered design or human-centered design in the human–computer interaction literature.

Figure 4.5. Threats and validation at each of the four levels. Many threats at the outer levels require downstream validation, which cannot be carried out until the inner levels within them are addressed, as shown by the red lines. Any single project would only address a subset of these levels, not all of them at once.
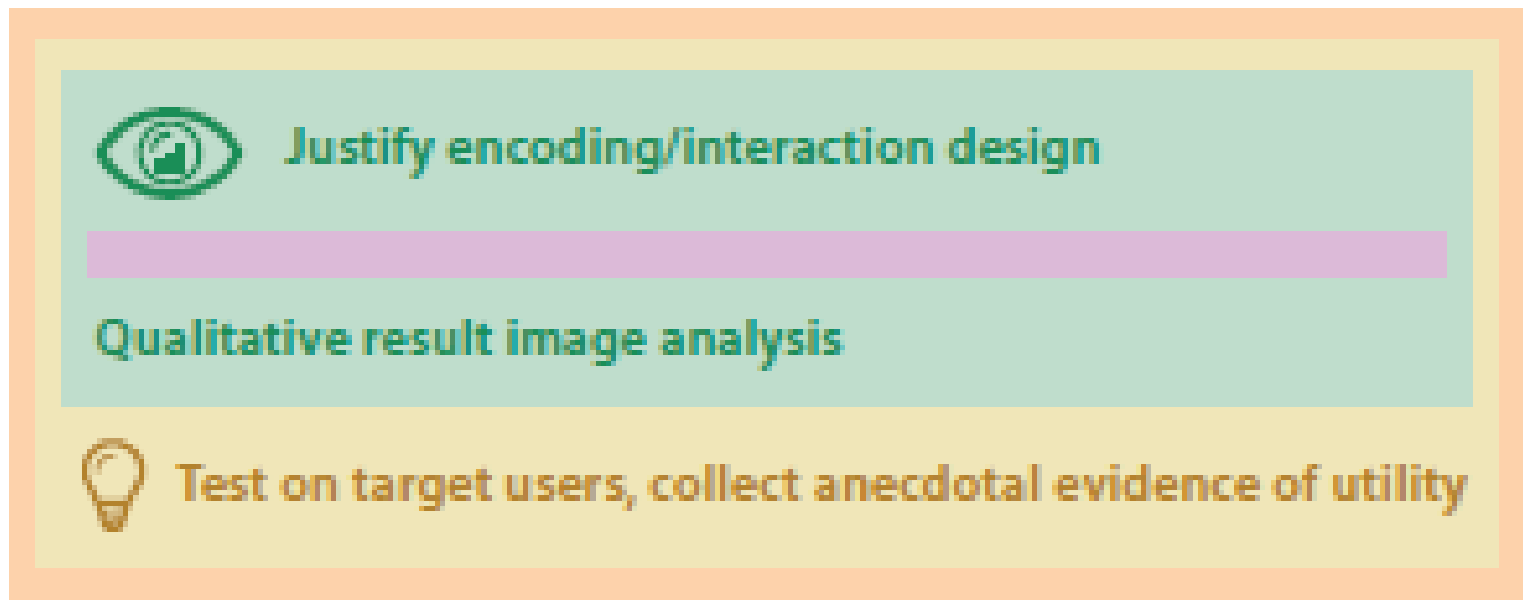

Threat Wrong problem
Validate Observe and interview target users

Threat Wrong task/data abstraction

Threat Ineffective encoding/interaction idiom
Validate Justify encoding/interaction design

Threat Slow algorithm
Validate Analyze computational complexity

Implement system

Validate Measure system time/memory

Validate Qualitative/quantitative result image analysis
Test on any users, informal usability study
Validate Lab study, measure human time/errors for task

Validate Test on target users, collect anecdotal evidence of utility
Validate Field study, document human usage of deployed system

Validate Observe adoption rates

Figure 4.6. Genealogical graphs. (a) Three layouts for the dual-tree: classical node–link top-to-bottom at the top, classical left-to-right on the left, and the new indented outline algorithm on the right. (b) Widget for subtree collapsing and expanding with ballistic drags. From [McGuffin and Balakrishnan 05, Figures 13 and 14].

Figure 4.7. Genealogical graphs [McGuffin and Balakrishnan 05] validation levels.
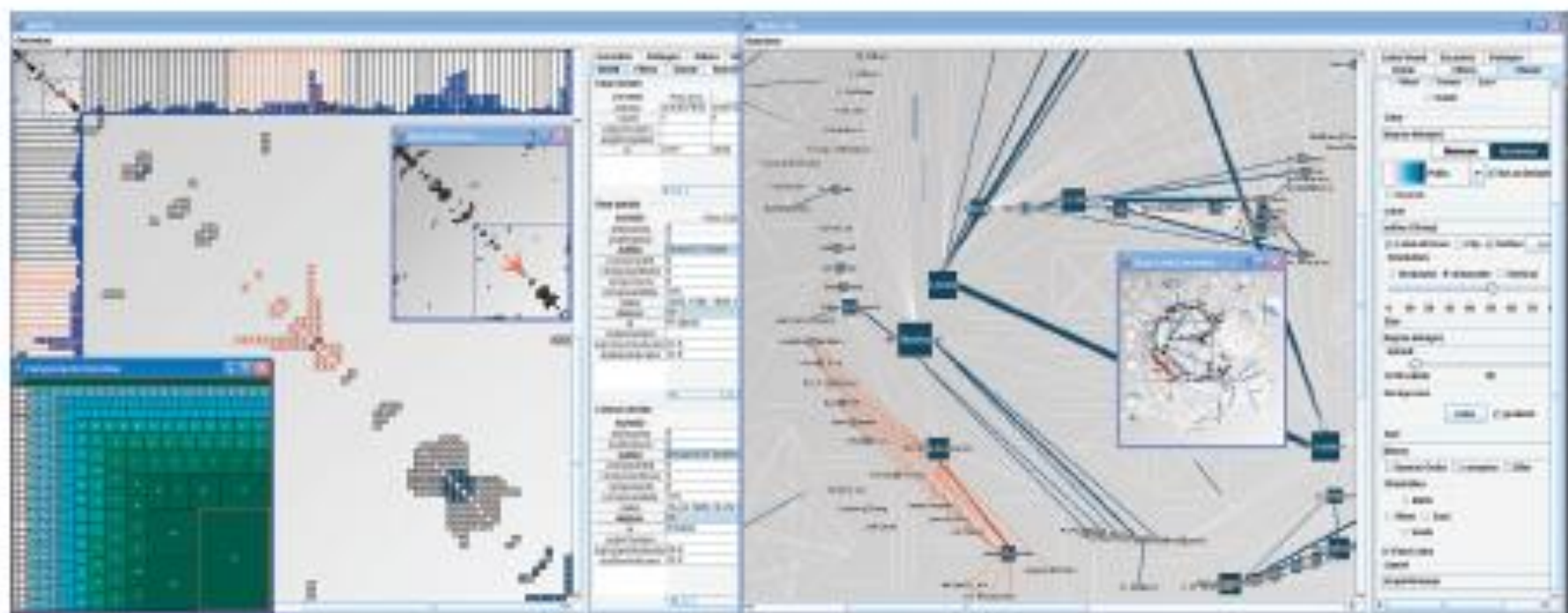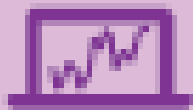
Figure 4.8. MatrixExplorer features both node–link and matrix representations in an interface designed for sociologists and historians to explore social networks. From [Henry and Fekete 06, Figure 1].

Figure 4.9. MatrixExplorer [Henry and Fekete 06] validation methods.