Promoting Open Science in Test-Driven Software Experiments

Marcus Kessel <marcus.kessel@uni-mannheim.de>, ORCID iD: 0000-0003-3088-2166

Invited Talk, Open Science Day 2024

About

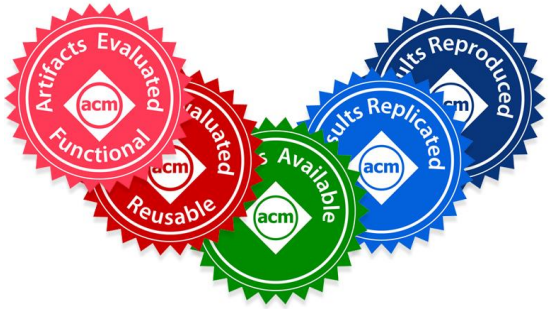
UNIVERSITÄT MANNHEIM

- Marcus Kessel, Postdoc (Software Engineering Group)
  - Doctors degree in Computer Science (Software Engineering)
  - https://www.wim.uni-mannheim.de/atkinson/team/dr-marcus-kessel/
- Research: Intersection of Software Engineering, Data Science and Experimentation
  - Interests
    - Program Analysis
    - Empirical Software Engineering and Measurement
    - Code Recommendation and Search
    - Mining Software Repositories (“Big Code”)
    - AI-powered Software (“AIWare”)
    - Open Science … and open software
  - Projects/Grants
    - *AUTOR* (Software Oracles, RiSC, MWK BW), *TDSEHub* (Shareable Studies, Open Science Grant)

Open Science in Software Engineering

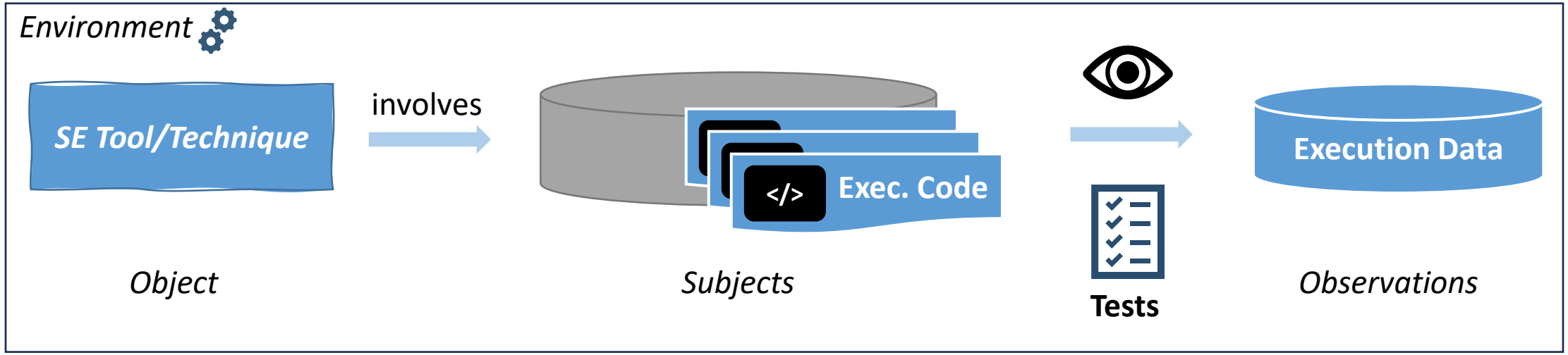
UNIVERSITÄT MANNHEIM

- The Gap between Ideal and Reality
  - Despite progress in the broader scientific community, the software engineering (SE) field lags behind in adopting open science principles
  - The SE "*community is struggling in adapting open science to the particularities of our discipline*" (Mendez et al.)
- Practice vs Principle
  - Leading SE conferences and journals emphasize empirical methods and reproducibility, but our community struggles to adapt these ideals to the unique needs of our discipline



https://www.acm.org/publications/artifacts

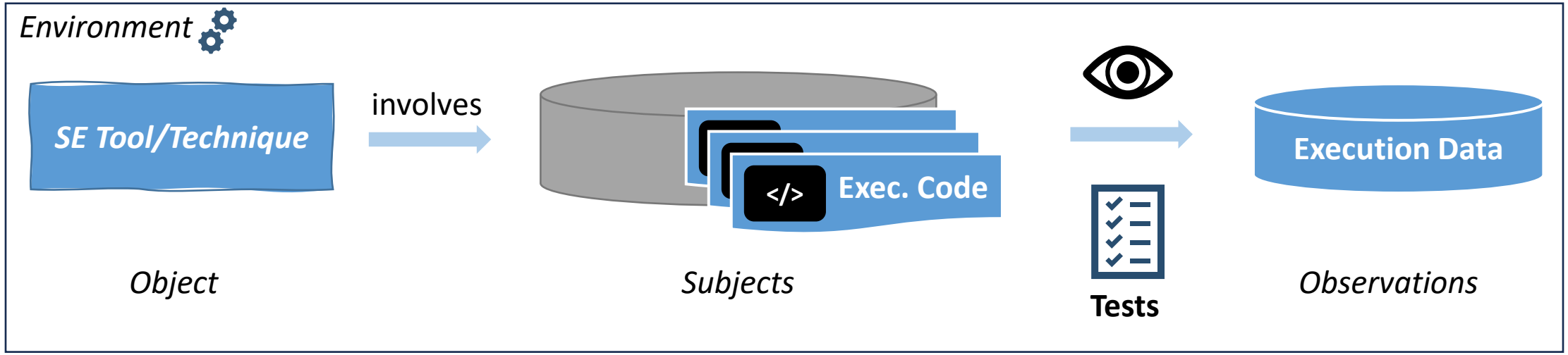
Test-driven Software Experiments (TDSEs)



Test-driven software experiments (TDSE)

- *"an experiment in which the experimental unit involves software (i.e., code) that is executed under controlled conditions by means of one or more tests"*
- Execution data: analysis of observational data obtained at run-time from code executions
- Example TDSEs
  - assessing performance of test generators
  - Benchmarking code LLMs for code generation tasks

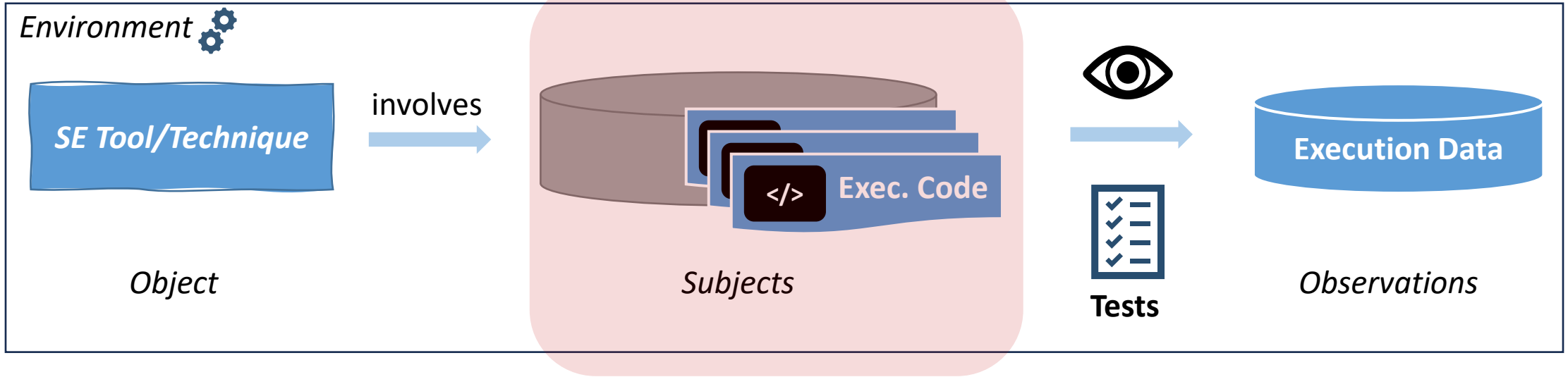
Problem



Reproducability of TDSEs

- Ad hoc representation of execution data makes it difficult to access and reuse
- Significant manual effort required to understand and utilize existing data
  - “here’s a link to GitHub” – that’s it
- Lack of standardization hinders the repetition, replication, and reproduction of TDSEs

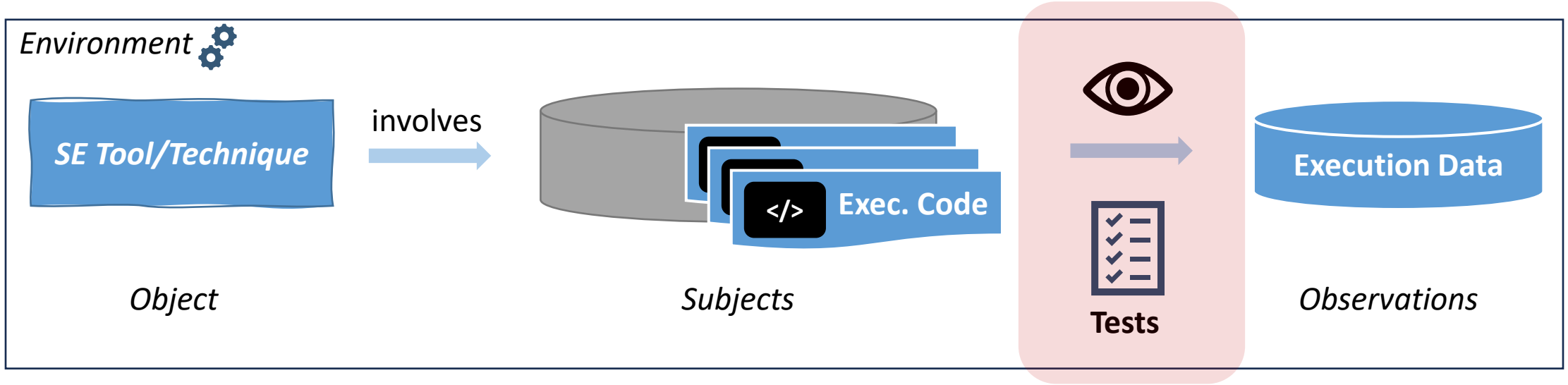
Challenges



Data Recoverability

- Lack of executable code corpora
- Often curated manually (tedious and error-prone) – hard to ensure executability
- Software data sets quickly degrade over time (maintenance costs etc.)

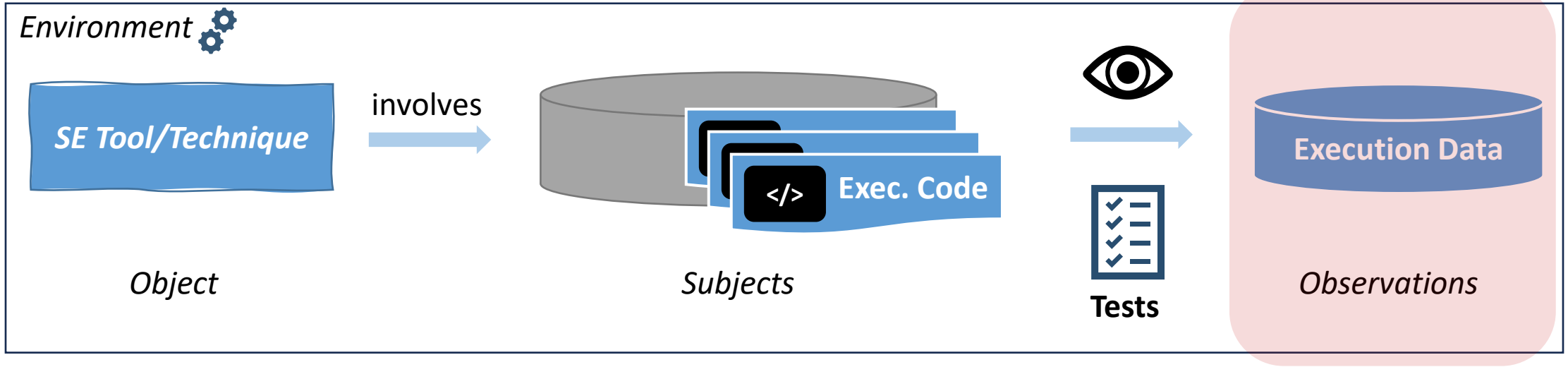
Challenges



Data Usability

- TDSEs require a lot of setup and adaptation code – often ad hoc solutions written by hand
- Poorly structured and documented
- Custom, ad hoc test specifications are hard to reuse and re-execute under the same environmental conditions

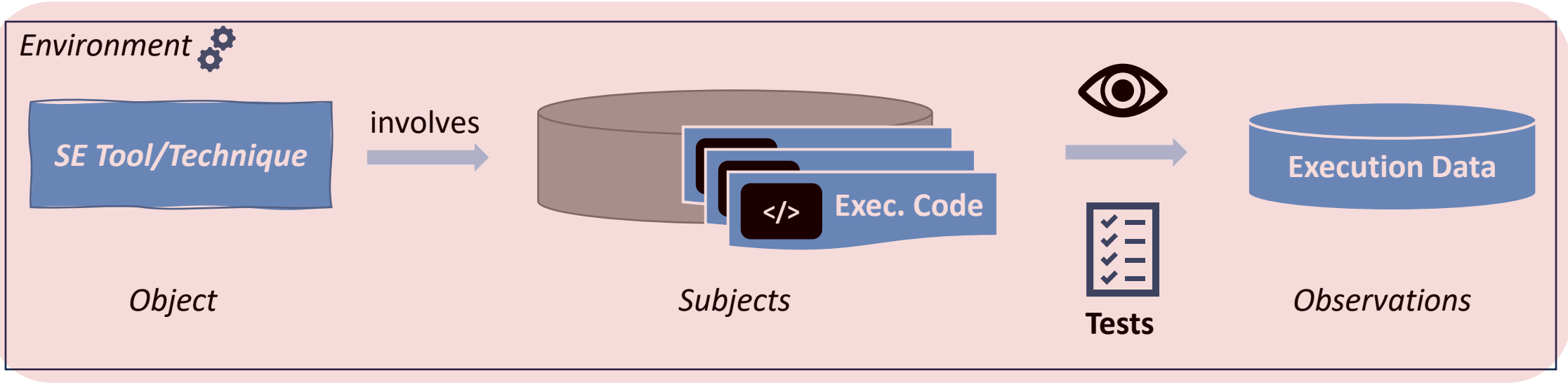
Challenges



Analytical Clarity

- No clear separation of concerns including the determination of important facets of test specifications (i.e., inputs and outputs)
- No systematic storage of execution data – often omitted and only final (aggregated) results reported
  - e.g., 100 tests passed – intransparent

Challenges



Consensual Results (Agreement)

- Analysis Pipeline: Study design encoded in ad hoc and non-standard ways – ad hoc scripting and custom code written for TDSEs in general-purpose languages
  - Often poorly documented
- Environment: Hard to configure the same settings and obtaining the same results
- Often intermediate steps and/or data required to reproduce them are omitted or produced manually by hand

Solution

- Large-scale Software Observatorium – LASSO
  - provides simple, transparent, domain-specific languages and data structures to create, analyze and store execution data at ultra-large scales

Executable Code Corpus

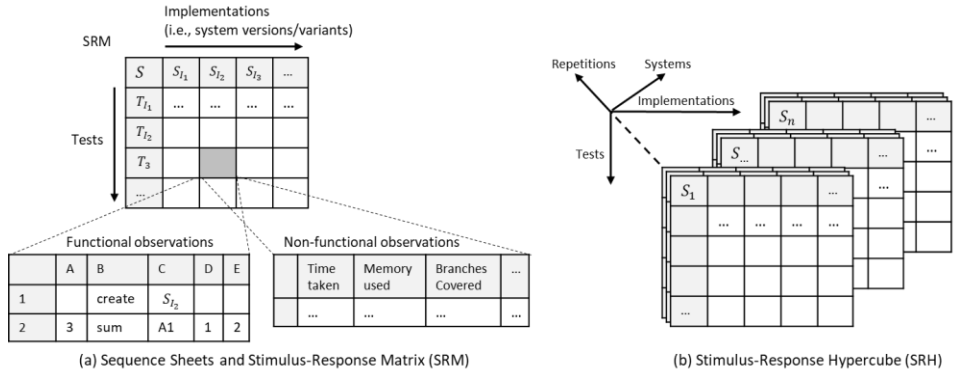
- extensible

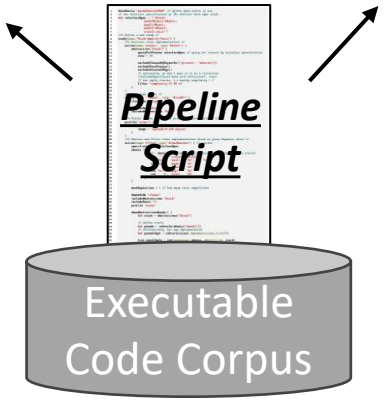
Reusable Data structures (tabular)

- Sequence Sheets (Test/Execution Data)
- Stimulus Response Matrices (SRMs)
- Stimulus Response Hypercubes (SRHs)

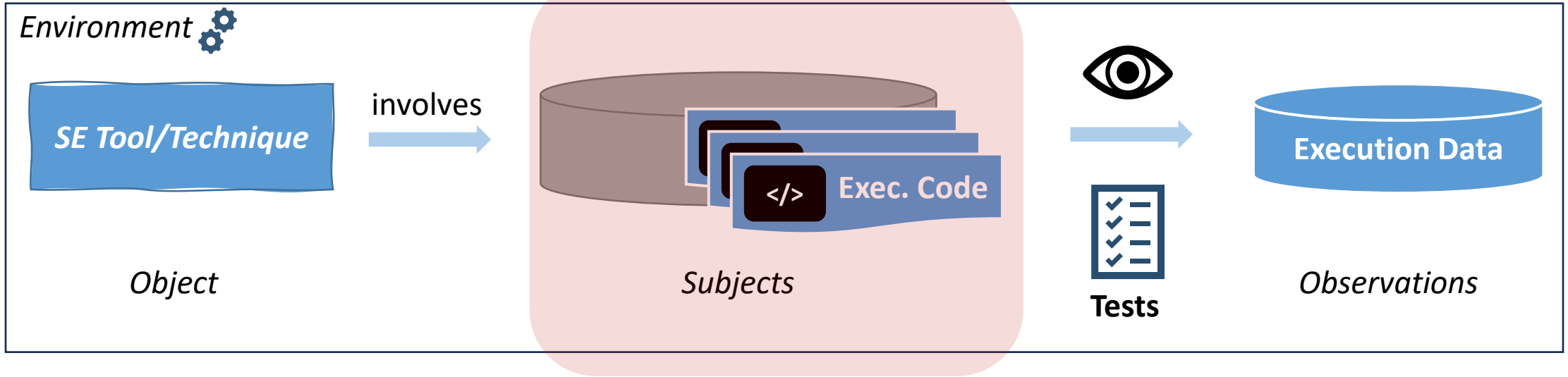
Domain-specific Languages

- Study designs: LASSO Scripting Language (LSL)
- Tests: Sequence Sheet Notation (SSN)





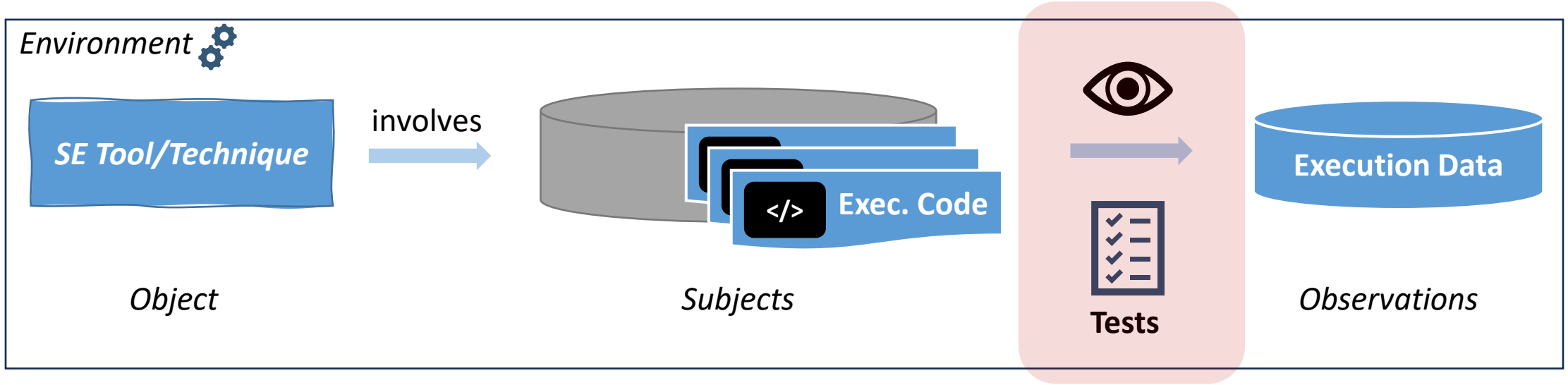
Solution



Data Recoverability

- Executable code corpus by LASSO
  - Guarantees executable code – automation
  - Enables automatic curation capabilities to create new TDSEs
  - Fosters accessibility and reusability
  - Extensible
    - Integrate your own datasets / benchmarks

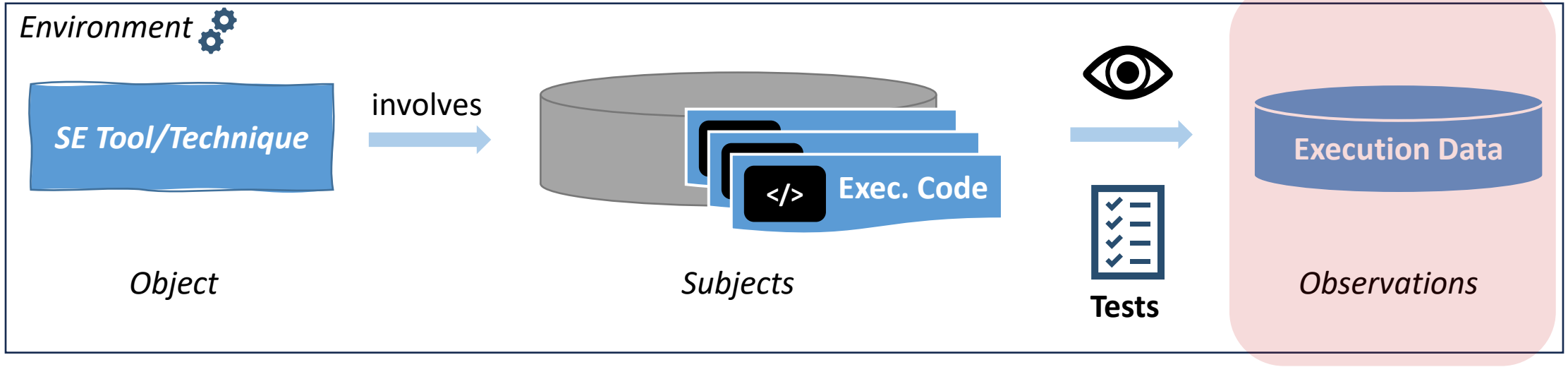
Solution



Data Usability

- Sequence Sheet Notation
  - reusable representation of tests well-suited for experimentation
  - tabular representations of run-time observations based on various dimensions
  - Interoperable format to facilitate analysis and comparisons of empirical results (e.g., comparing two experiments)

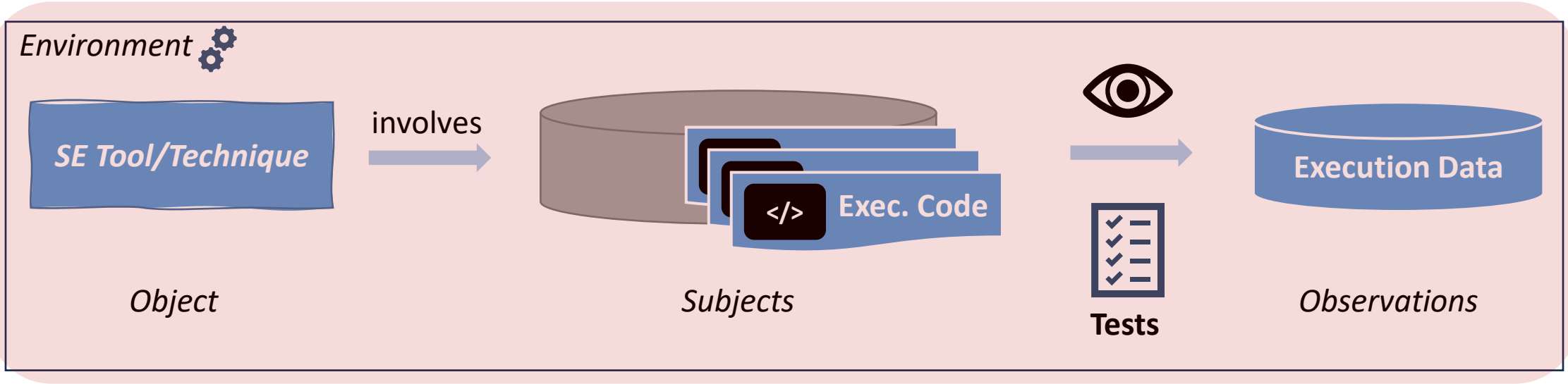
Solution



Analytical Clarity

- Data Structures – SRMs and SRHs
  - Systematic storage of all execution data – tabular representation (interoperability)
  - Retaining explicit links to executed code and tests
  - Clear separation of concerns (inputs, operations, output)
  - Stores any observational types (functional and non-functional observations)

Solution



Consensual Results (Agreement)

- Lasso Query Language (LSL)
  - Offers executable study designs written in a domain-specific language
  - Scripts (and results) can be shared and re-executed to enable reproducibility
  - LASSO serves as a general-purpose platform that offers the data structures and languages
  - SRMs produced by TDSEs can be effciently compared

Demonstration

UNIVERSITÄT MANNHEIM

- Successfully replicated a recent benchmark for code LLMs based on 4 Rs

<table>
<tr><th rowspan="2">Purpose/ Design
Team</th><th colspan="2">Same Purpose (i.e., RQs)</th><th>Different Purpose (i.e., RQs)</th></tr>
<tr><th>Same Design</th><th>Different Design</th><th rowspan="3">Repurposing:
• different purpose (i.e., RQs)
• original or independent team
• different (logical) design
• performed in a different way</th></tr>
<tr><td>Original Team</td><td>Repetition:
• same purpose
• original team
• same (logical) design</td><td rowspan="2">Reproduction:
• same purpose
• original or independent team
• different (logical) design</td></tr>
<tr><td>Different Team</td><td>Replication:
• same purpose
• independent team
• same (logical) design</td></tr>
</table>

Conclusion and Future Work

UNIVERSITÄT MANNHEIM

- **Simplifying TDSEs with LASSO**: We've introduced novel data structures and languages, empowering researchers to tackle complex challenges in a more streamlined way
- **Fostering Open Science Practices**: By promoting the sharing, reuse, and adaptation of TDSEs, we're enabling the scientific community to accelerate progress through open science principles.
- **Expanding the Impact of LASSO**: We invite third-party developers to leverage our data structures and languages in their own platforms, further amplifying the reach and utility of LASSO
- Funding opportunities will support continued innovation and community growth around LASSO
  - Open Science Grant (2024-2025) - **TDSE-Hub: A Repository for Reproducible Test-driven Software Experiments**
    - This dedicated platform will serve as a central hub for sharing, accessing, and reusing TDSEs, promoting reproducibility and transparency in test-driven software experiments

UNIVERSITÄT MANNHEIM

Thank you!

Resources

UNIVERSITÄT MANNHEIM

- Kessel, Marcus, and Colin Atkinson. "Promoting open science in test-driven software experiments." Journal of Systems and Software 212 (2024):111971. https://doi.org/10.1016/j.jss.2024.111971
  - Open Access
- LASSO Platform, https://softwareobservatorium.github.io/
  - Open Source