Mining Software Repositories for Prediction

Software Evolution — L6T3

Dr. Vadim Zaytsev aka @grammarware, March 2021

UNIVERSITY OF TWENTE. | Formal Methods & Tools

Why

- Older than dirt question:
  - when to stop testing? when is enough?
- More broadly:
  - what is the most efficient QA resource allocation?
- Some modules are more error-prone than others
- Issue trackers are a big help

What Makes a Module Error Prone?

- Complexity
- Integration
- Problem domain
- Requirements creep
- Process

Experimental Setup

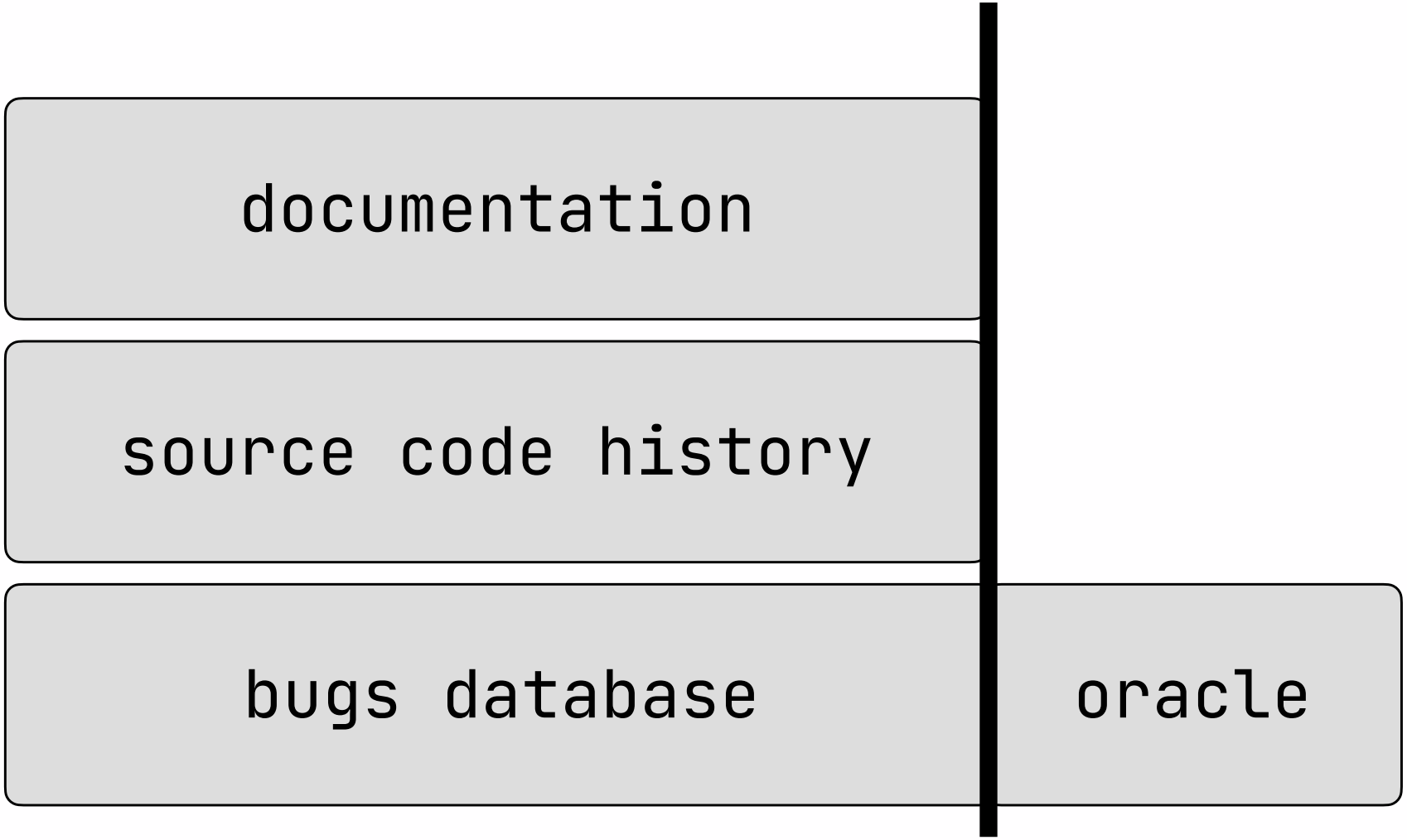
- Split
- Precision
- Recall
- F-score
- Correlation
- Co-occurrence

Main Approaches

- Change Log
  - assumption: recently/often changed files = bugs
- Single Version
  - assumption: current design/behaviour ⇒ defects
- Dependencies
  - assumption: cohesion ~ vulnerable

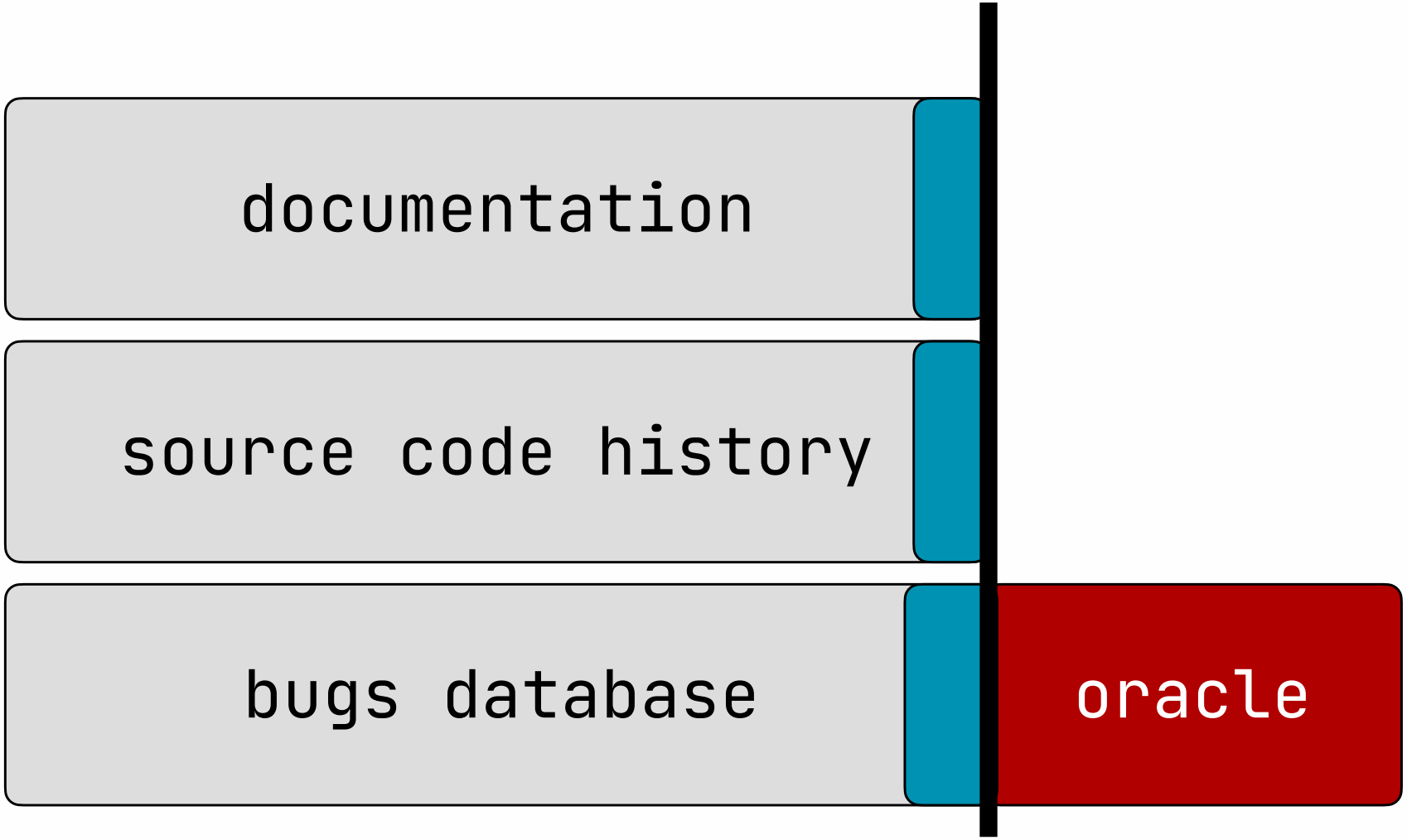
D'Ambros, Lanza, Robbes, *An Extensive Comparison of Bug Prediction Approaches*, MSR 2010

Bug Prediction & Data Sources



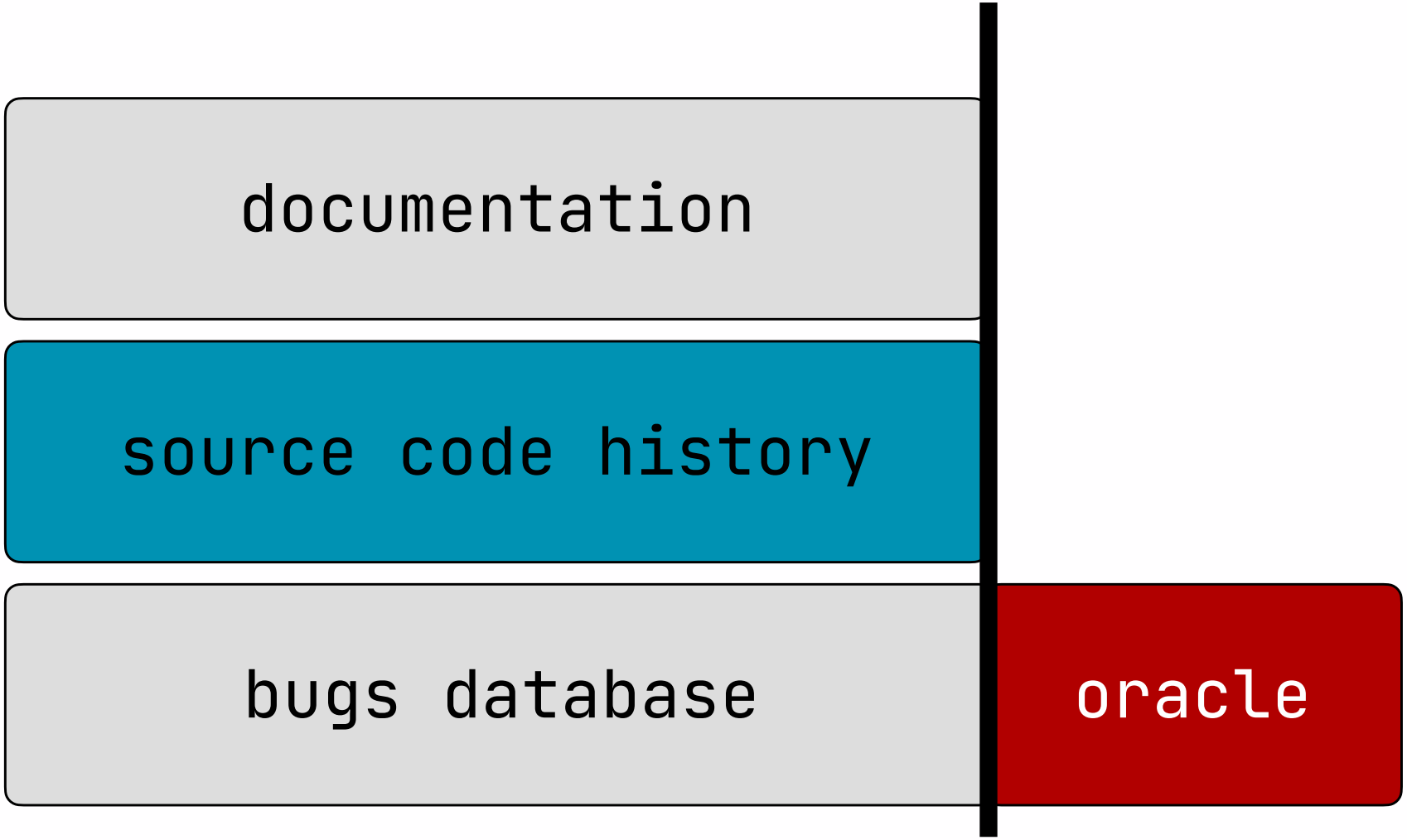
D'Ambros, Lanza, Robbes, *An Extensive Comparison of Bug Prediction Approaches*, MSR 2010

Bug Prediction & Data Sources



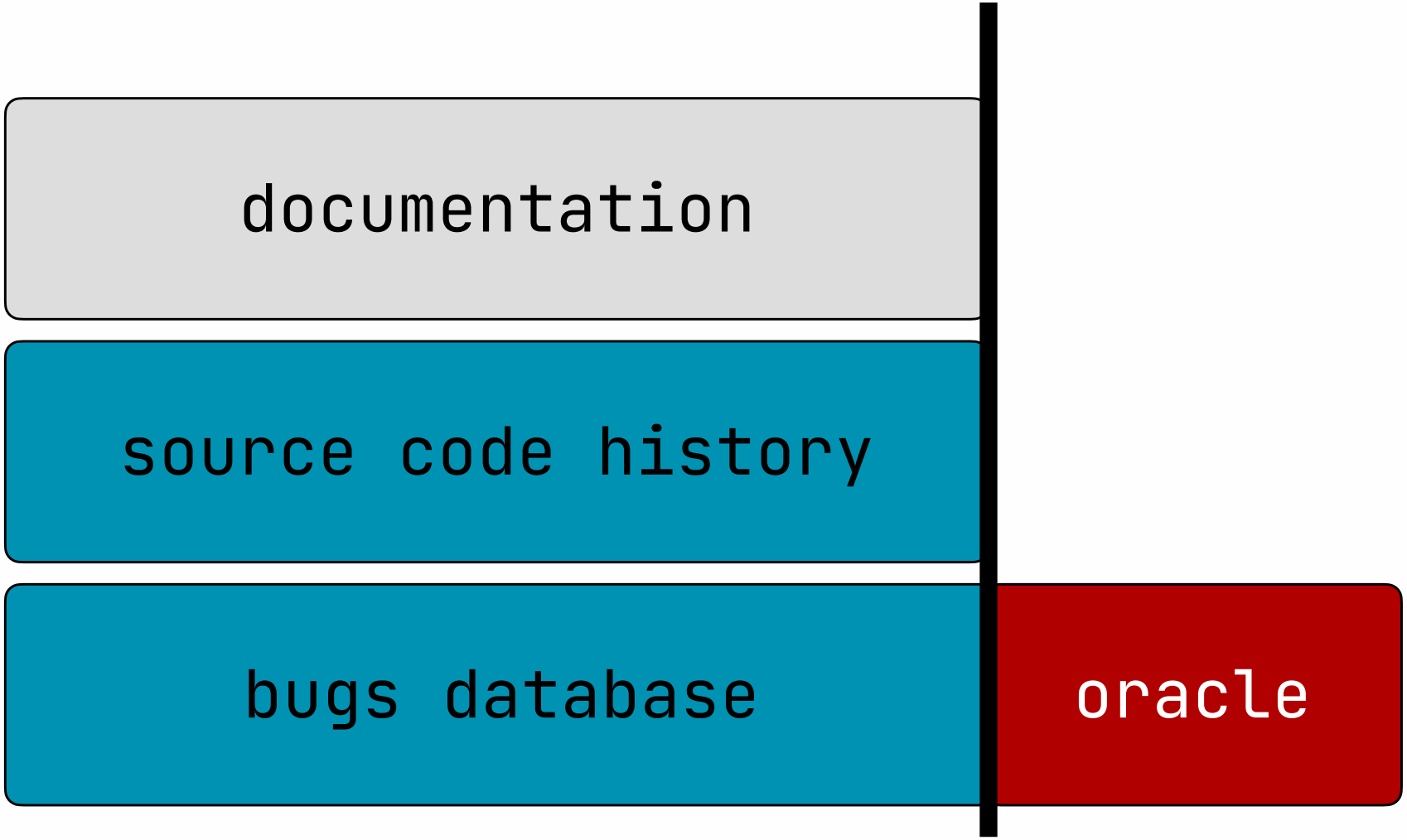
D'Ambros, Lanza, Robbes, *An Extensive Comparison of Bug Prediction Approaches*, MSR 2010

Bug Prediction & Data Sources



D'Ambros, Lanza, Robbes, *An Extensive Comparison of Bug Prediction Approaches*, MSR 2010

Bug Prediction & Data Sources



D'Ambros, Lanza, Robbes, *An Extensive Comparison of Bug Prediction Approaches*, MSR 2010

Conclusion

- Learning from the past matters for the future
- There is a place for empirical SE
- Data points: bugs, code, docs
- Q&A Sessions @ Canvas
  - ⇒ v.zaytsev@utwente.nl
  - ⇒ https://discord.gg/n7VQAPNBPD