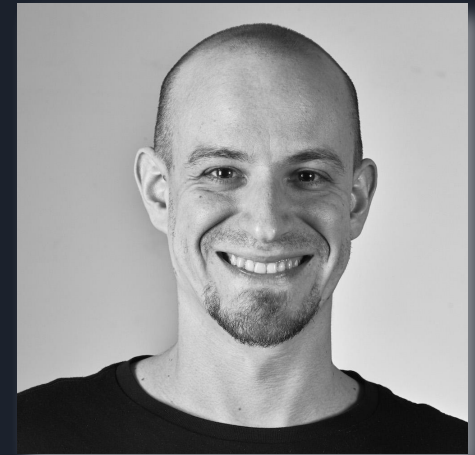# Architectural Technical Debt Identification: *Moving Forward*

Roberto Verdecchia

r.verdecchia@vu.nl

About me

- PhD student at:
  - *Gran Sasso Science Institute*
  - *Vrije Universiteit Amsterdam*

- Academic age: **1.5 years**

- Total Ph.D. duration: **3 years**

- Year 1: courses software engineering, algorithms and formal methods

- web: roberto.verdecchia.github.io

Architectural Technical Debt

- **Sub-optimal decisions** resulting in **immature architectural artifacts**[1]

- Architectural Technical Debt Items (**ATDIs**) have potentially **high impact** on overall TD

- Hard to uncover

- Identifying ATDIs is still an **open problem**

[1] "A Systematic Literature Review and a Unified Model of ATD." IEEE, Aug. 2016, pp. 189–197. T. Besker, A. Martini, and J. Bosch

(Ultimate)

Research Goal

Understand how to efficiently and effectively identify ATDIs present in software-intensive systems

Research questions

RQ1: Do **modification summaries**, **commit log messages**, **issue trackers**, etc. provide more ATD information than code alone?

RQ2: Which ATDI can be identified **automatically** from artifacts of version repositories?

RQ3: Which ATDI tend to require **additional human input** to be identified?

Methodology (bis)

- Evaluation through **empirical experiments**

- **OSS** & **Industrial** case studies (pros / cons)

- Mix of **quantitative** & **qualitative** analysis

Feedback (keep it coming!)

How to recognize that self-admitted TD is "architectural"?

What about architectural antipatterns?

Which esperimental subjects?

What to do after ATDIs Are identified?

Focus on specific types of ATDIs!

Think about validation!

Doesn't identification always imply ranking?

What would be the Minimal input set?

(Ultimate) Research Goal

Understand how to efficiently and effectively identify ATDIs present in software-intensive systems
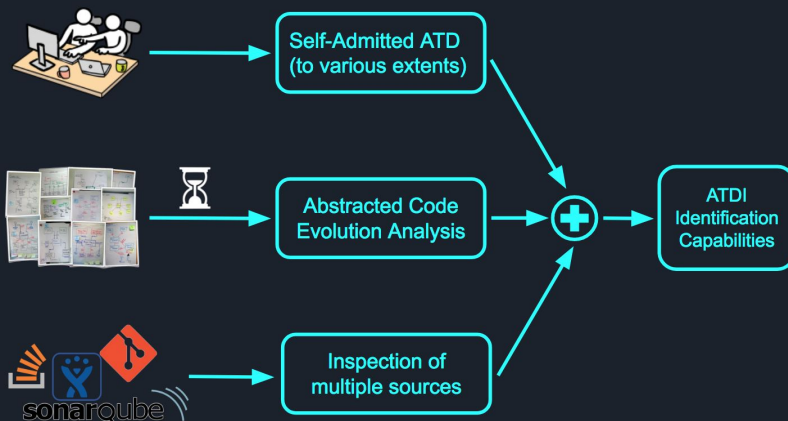
Research questions

*RQ1: Do **modification summaries**, **commit log messages**, **issue trackers**, etc. provide more ATD information than code alone?*

*RQ2: Which ATDI can be identified **automatically** from artifacts of version repositories?*

*RQ3: Which ATDI tend to require **additional human input** to be identified?*

Methodology

Self-Admitted ATD (to various extents)

Abstracted Code Evolution Analysis

Inspection of multiple sources

sonarqube

ATDI Identification Capabilities

Feedback (keep it coming!)

How to recognize that self-admitted TD is "architectural"?

What about architectural antipatterns?

Which esperimental subjects?

What to do after ATDIs are identified?

Focus on specific types of ATDIs!

What would be the minimal input set?

Think about validation!

Doesn't identification always imply ranking?

Architectural Technical Debt Identification: *Moving Forward*

Roberto Verdecchia

r.verdecchia@vu.nl