

Seattle, 1 May 2018

Architectural Technical Debt Identification: *Moving Forward*

Roberto Verdecchia

roberto.verdecchia@gssi.it



Architectural Technical Debt

- **Sub-optimal decisions** resulting in the conceivment of **immature architectural artifacts**¹
- Architectural Technical Debt Items (**ATDIs**) have potentially **high impact** on overall TD
- Hard to undercover
- Accurately 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



Methodology



Self-Admitted ATD
(to various extents)

Methodology



Self-Admitted ATD
(to various extents)



Methodology



Self-Admitted ATD
(to various extents)



Abstracted Code
Evolution Analysis

Methodology



Self-Admitted ATD
(to various extents)

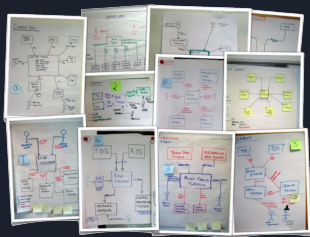


Abstracted Code
Evolution Analysis

Methodology



Self-Admitted ATD
(to various extents)



Abstracted Code
Evolution Analysis



Inspection of
multiple sources

Methodology



Self-Admitted ATD
(to various extents)



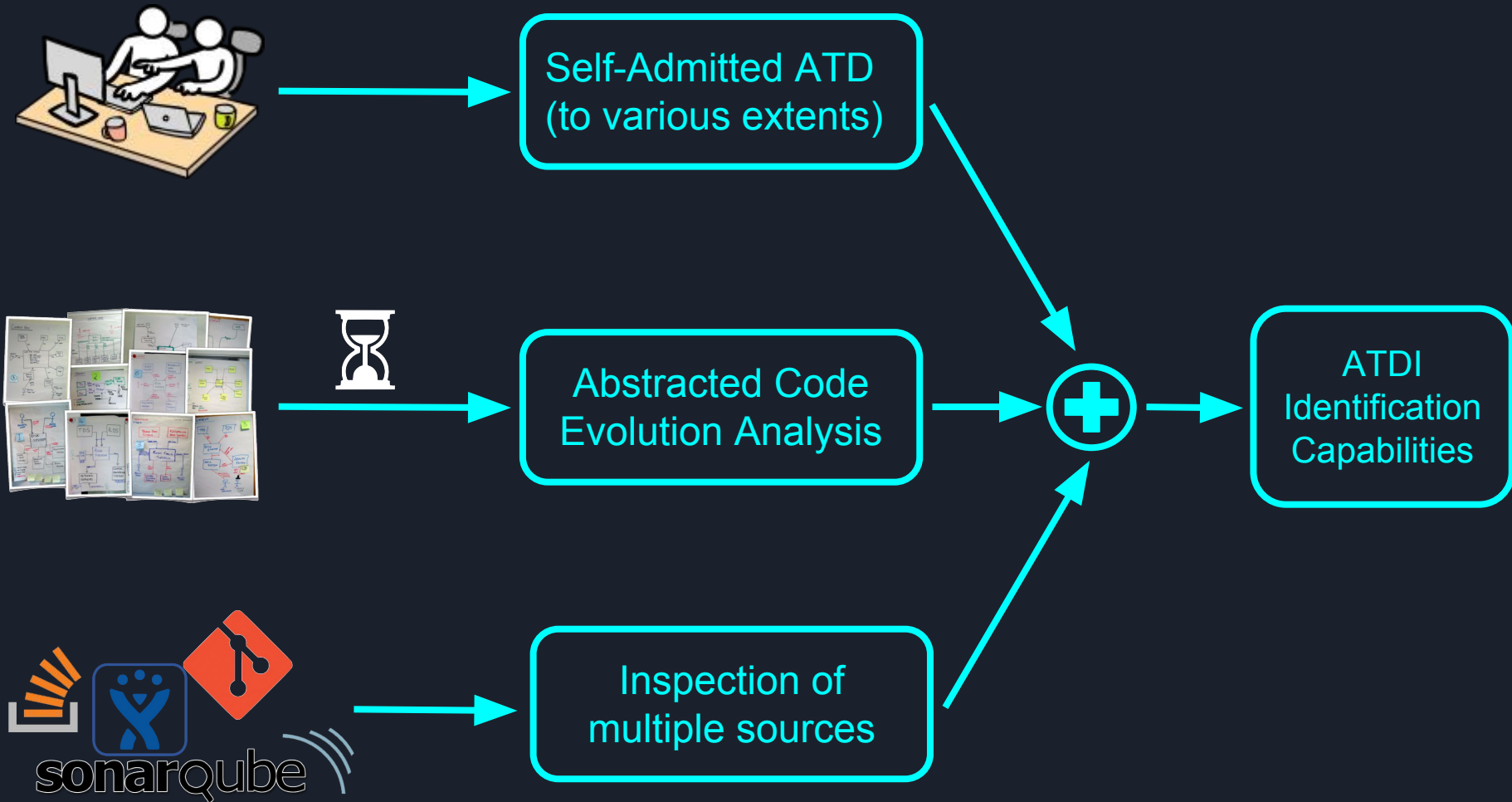
Abstracted Code
Evolution Analysis



Inspection of
multiple sources



Methodology





Methodology (bis)

- Evaluation through **empirical experiments**
- **OSS & Industrial** case studies (pros / cons)
- Mix of **quantitative & qualitative** analysis

(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

