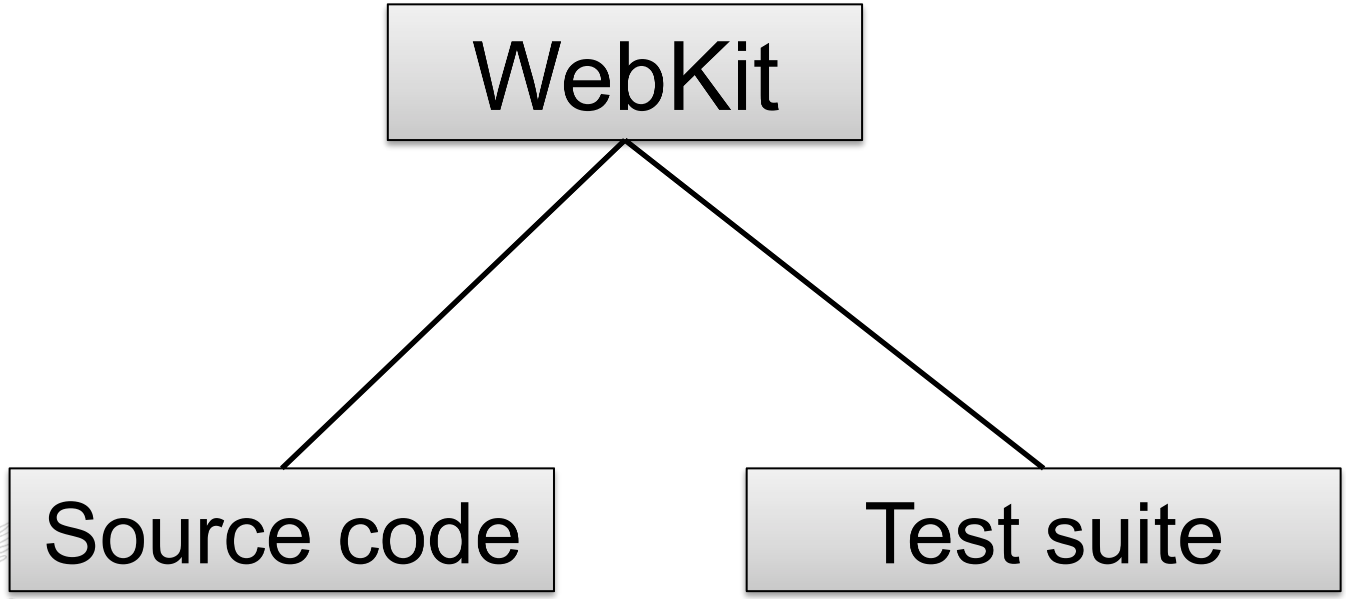
Beyond Code Coverage – an Approach for Test Suite Assessment and Improvement

Dávid Tengeri*, *Árpád Beszédes**,*
*Tamás Gergely**, *László Vidács*❖*,*
Dávid Havas and Tibor Gyimóthy**

**Department of Software Engineering, University of Szeged, Hungary*
❖*MTA-SZTE Research Group on Artificial Intelligence, University of Szeged, Hungary*

Industrial Motivation

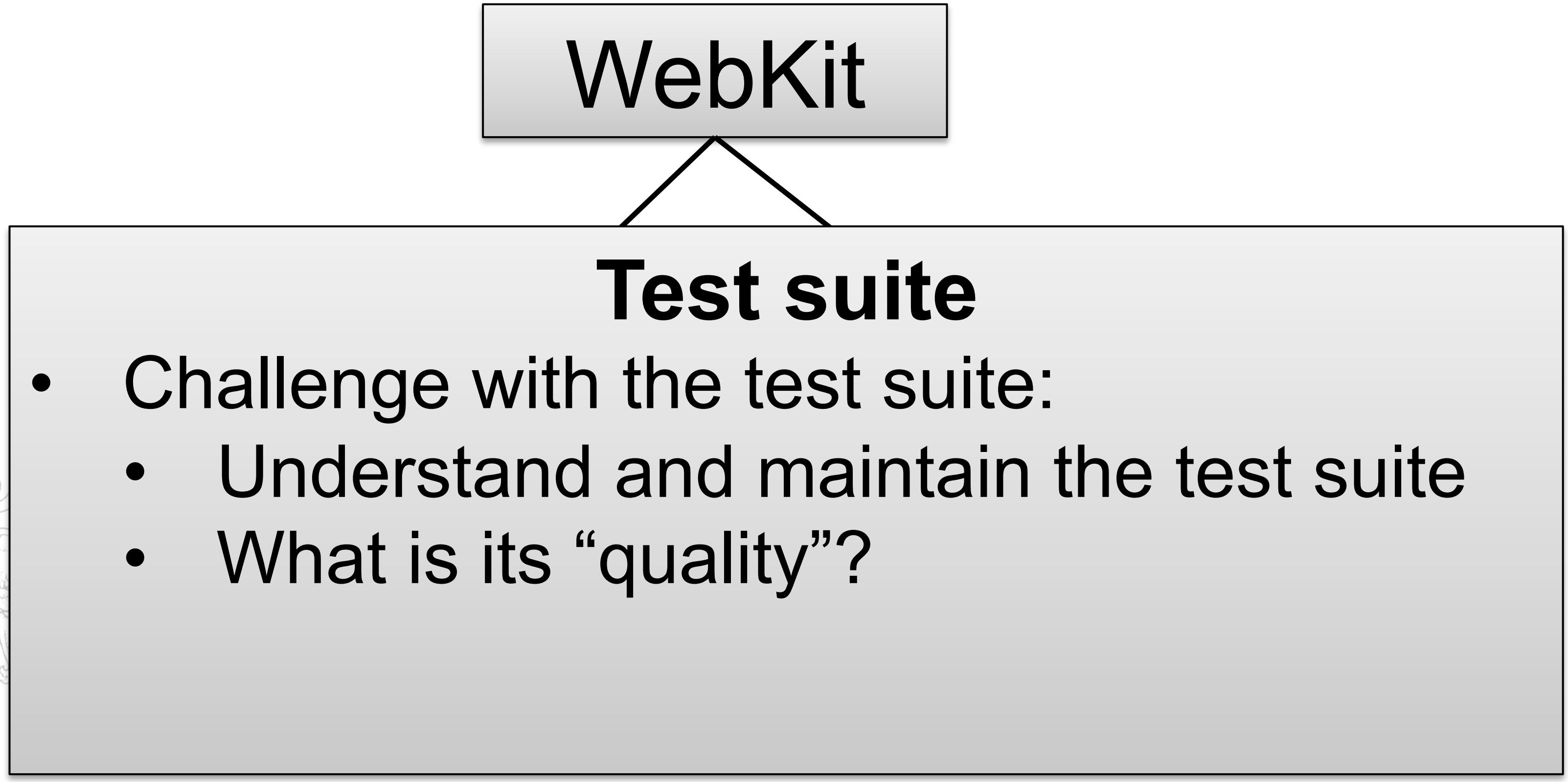


Industrial Motivation

WebKit

- Open source web browser engine
- ~ 2.2 million lines of code (mostly C++)
- More than 27 000 test cases

Industrial Motivation



WebKit

Test suite

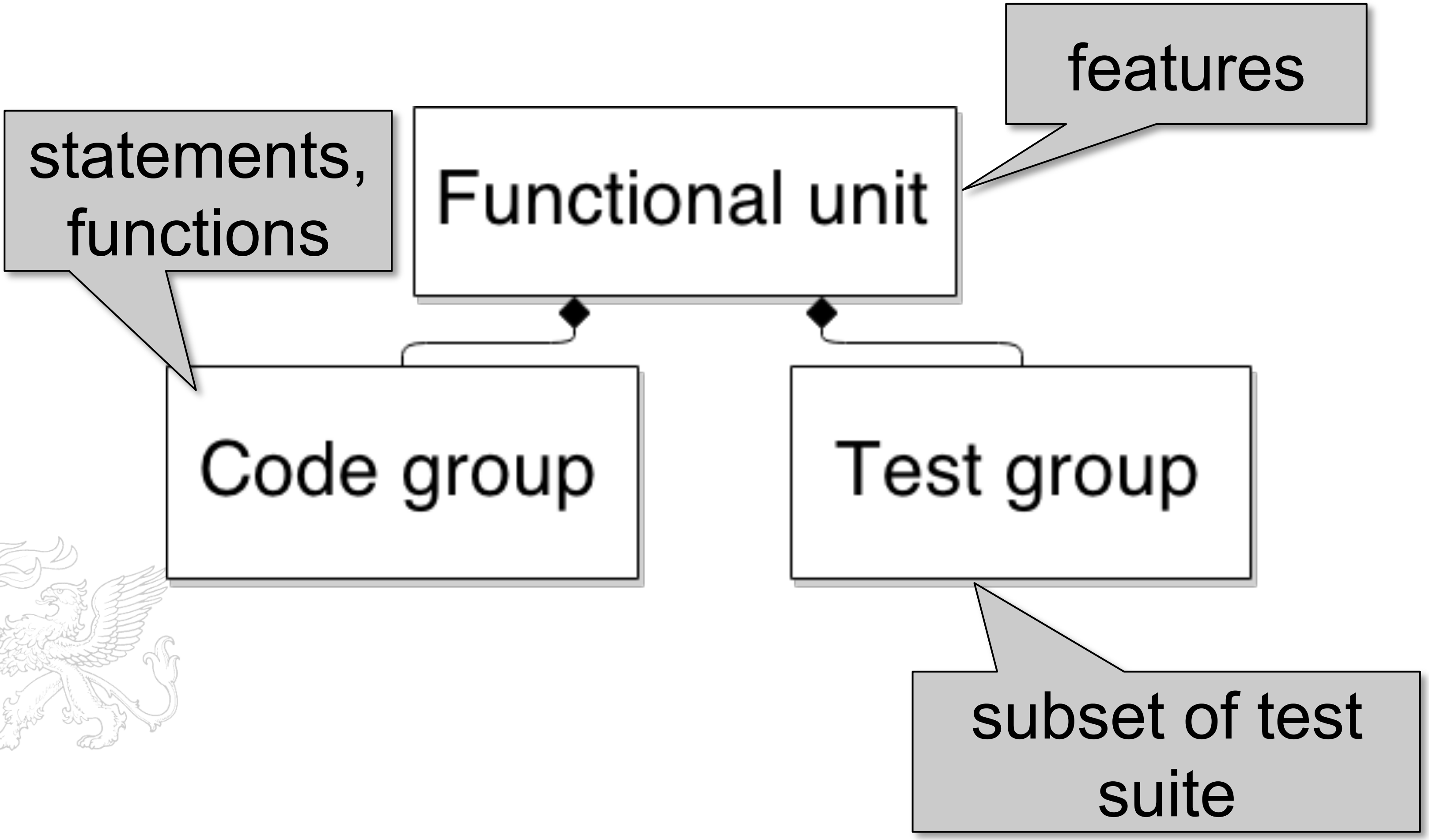
- Challenge with the test suite:
  - Understand and maintain the test suite
  - What is its “quality”?

Motivation and Goals

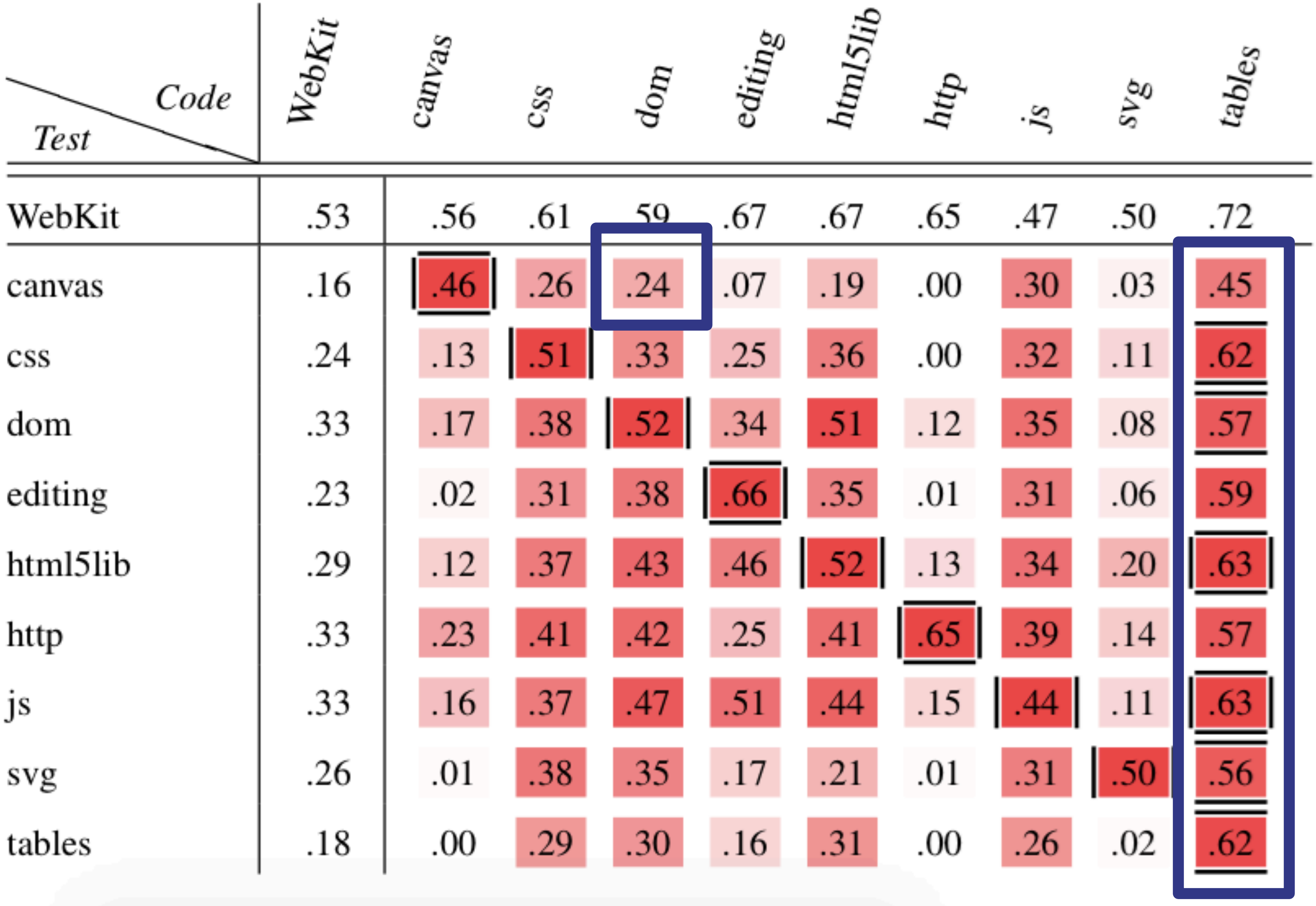
- “Quality” of a (regression) test suite:
  - What is the likelihood of defect detection?
  - How efficient, modular, etc. it is?
- How can the quality be improved?
- Quality assessment of source code – problem solved (?)
- Quality assessment of ***test suites***?

Test suite Assessment and Improvement Method (TAIME)

Determine functional units



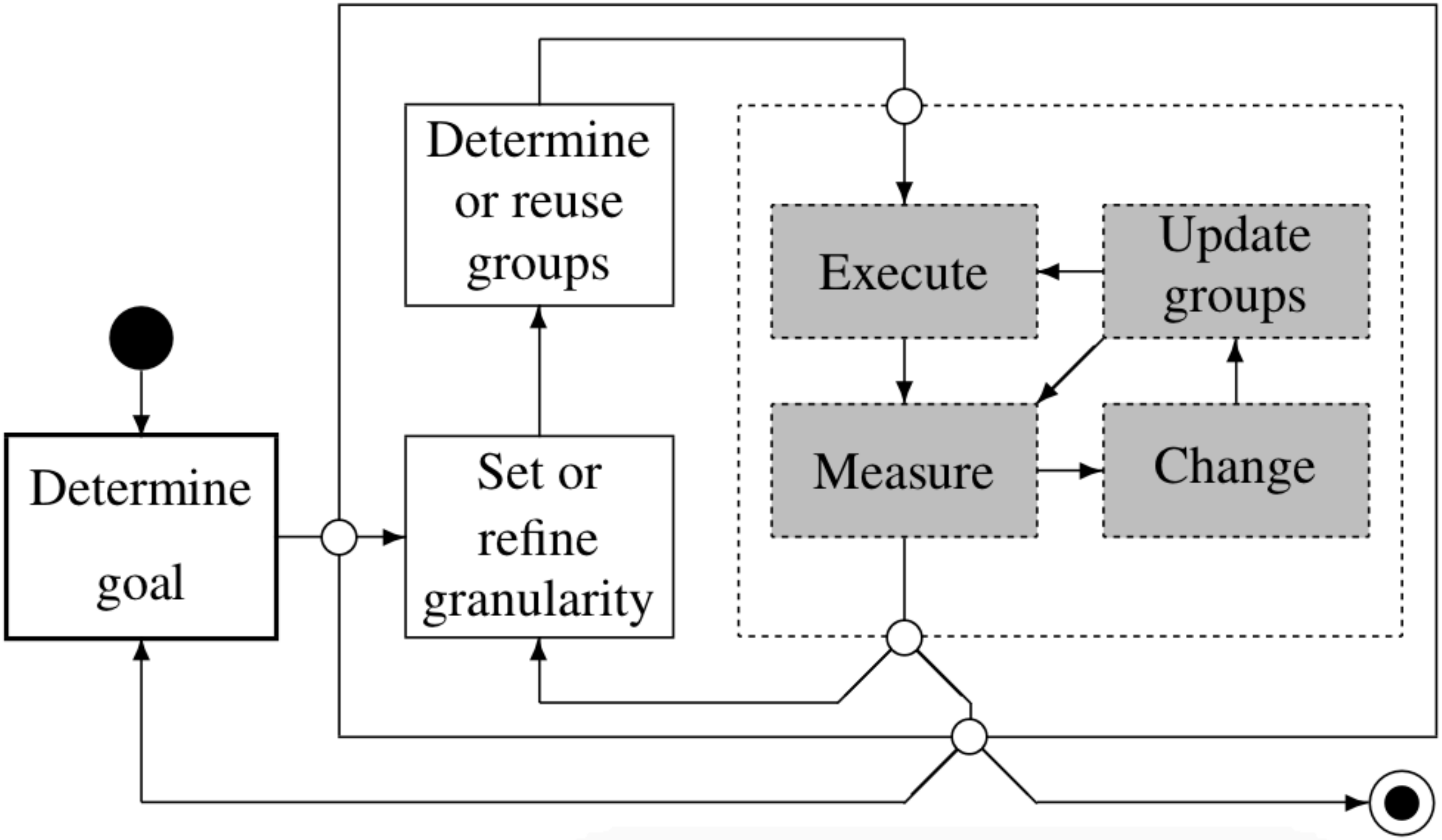
Assessment of WebKit



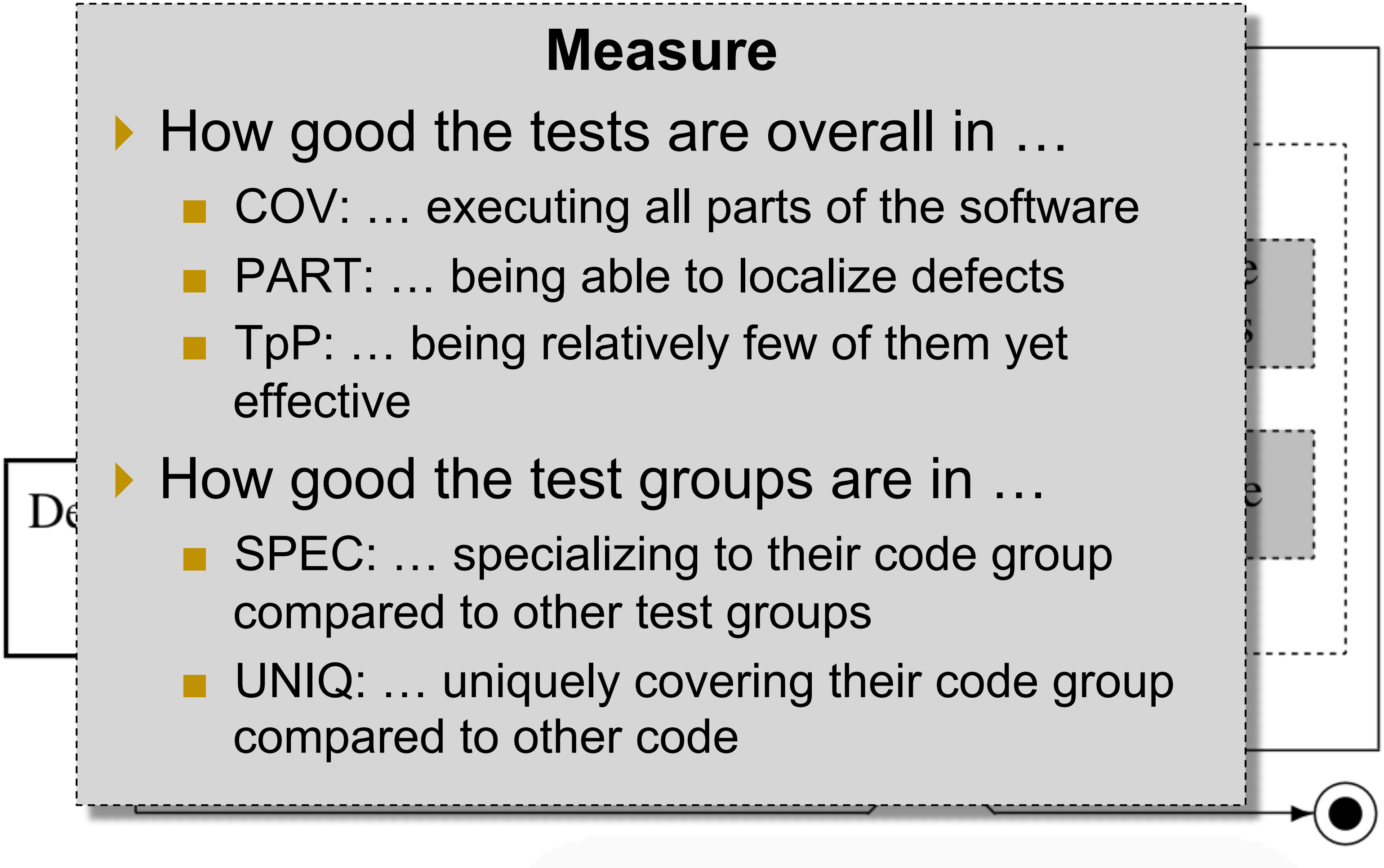
Code / Test	WebKit	canvas	css	dom	editing	html5lib	http	js	svg	tables
WebKit	.53	.56	.61	.59	.67	.67	.65	.47	.50	.72
canvas	.16	.46	.26	.24	.07	.19	.00	.30	.03	.45
css	.24	.13	.51	.33	.25	.36	.00	.32	.11	.62
dom	.33	.17	.38	.52	.34	.51	.12	.35	.08	.57
editing	.23	.02	.31	.38	.66	.35	.01	.31	.06	.59
html5lib	.29	.12	.37	.43	.46	.52	.13	.34	.20	.63
http	.33	.23	.41	.42	.25	.41	.65	.39	.14	.57
js	.33	.16	.37	.47	.51	.44	.15	.44	.11	.63
svg	.26	.01	.38	.35	.17	.21	.01	.31	.50	.56
tables	.18	.00	.29	.30	.16	.31	.00	.26	.02	.62

Function level coverage of groups in WebKit

The TAIME approach



The TAIME approach



Measure

- How good the tests are overall in …
  - COV: … executing all parts of the software
  - PART: … being able to localize defects
  - TpP: … being relatively few of them yet effective
- How good the test groups are in …
  - SPEC: … specializing to their code group compared to other test groups
  - UNIQ: … uniquely covering their code group compared to other code

Use cases

Assessment

- Detect any issues that require further investigation
- Helps find the initial goal in the improvement phases

One-shot refactoring

- Delete, reorganize, rewrite, create test cases
- To improve test suite quality

Change-oriented test suite evolution

- Create new test cases or possibly remove test cases
- To preserve or improve test suite quality

White-box test design

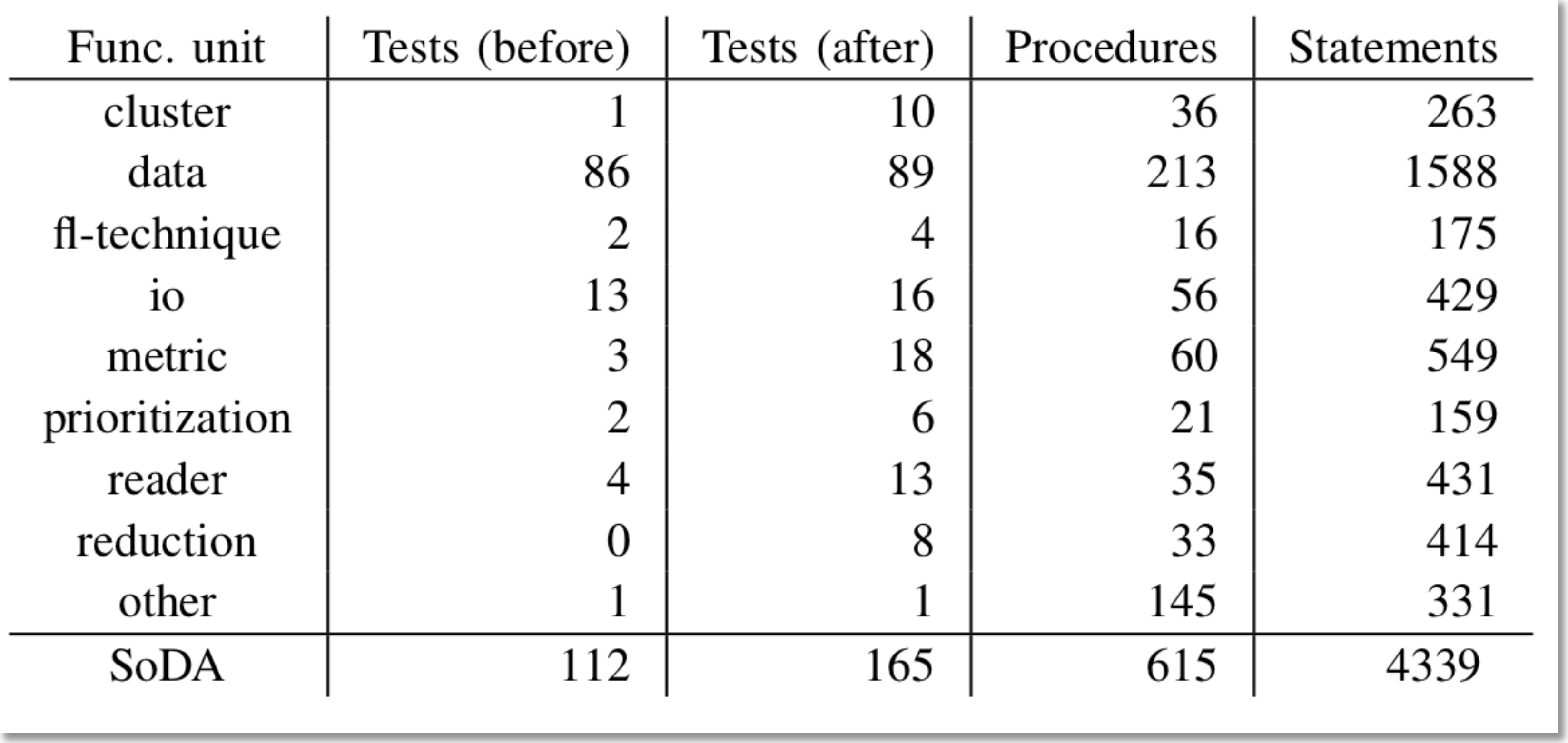
- Monitor and maintain test suite quality during design
- Use specific measurements as white-box criteria

Improvement of SoDA

- Software Development Analysis Framework
  - http://soda.sed.hu
  - Platform independent
  - Plugin based
  - TAIME support (with GUI)
- SoDA Repository
  - Benchmark programs (SIR, WebKit, GCC)
  - Set of measurement results

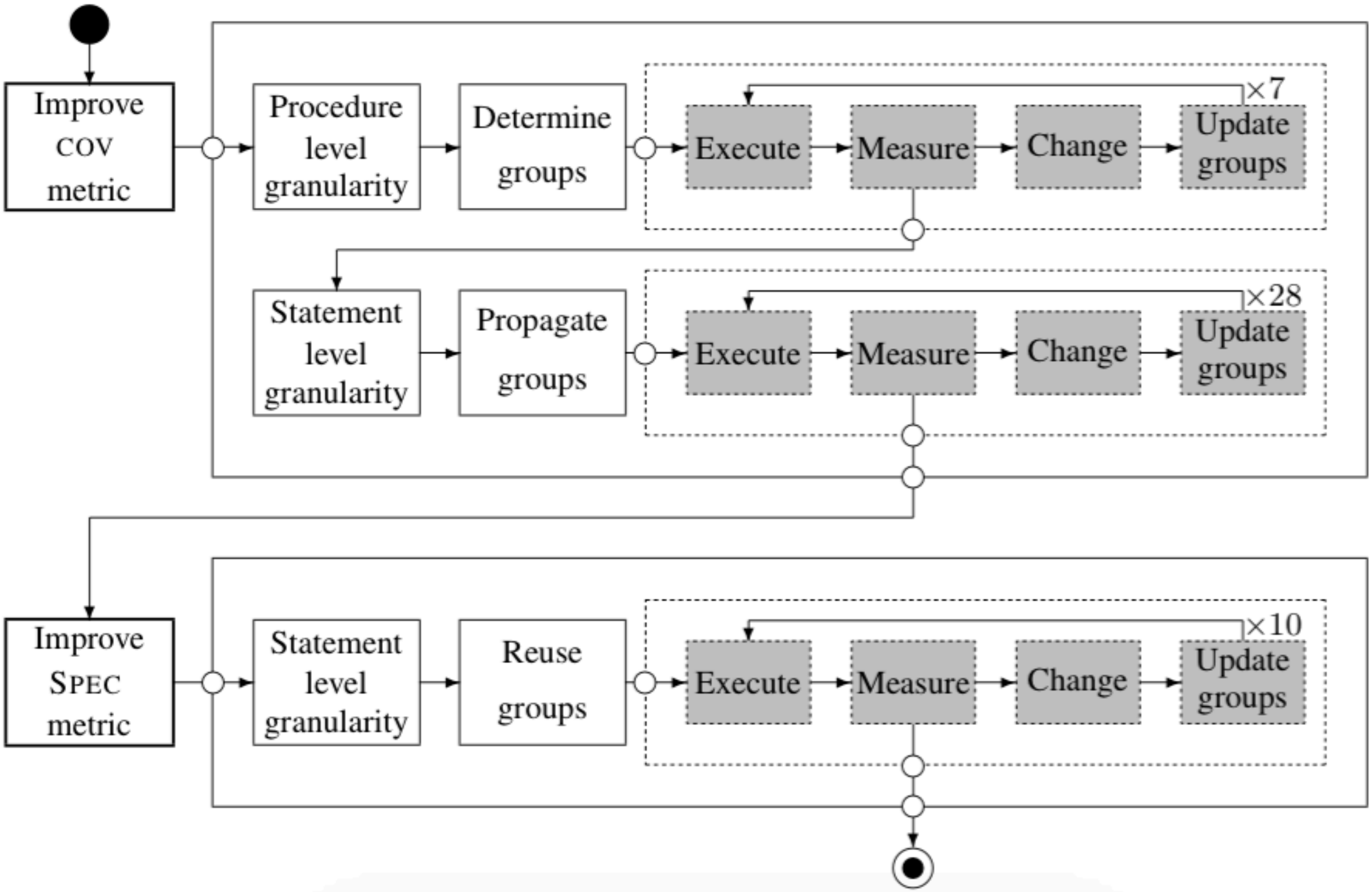
Improvement of SoDA

- White-box test design

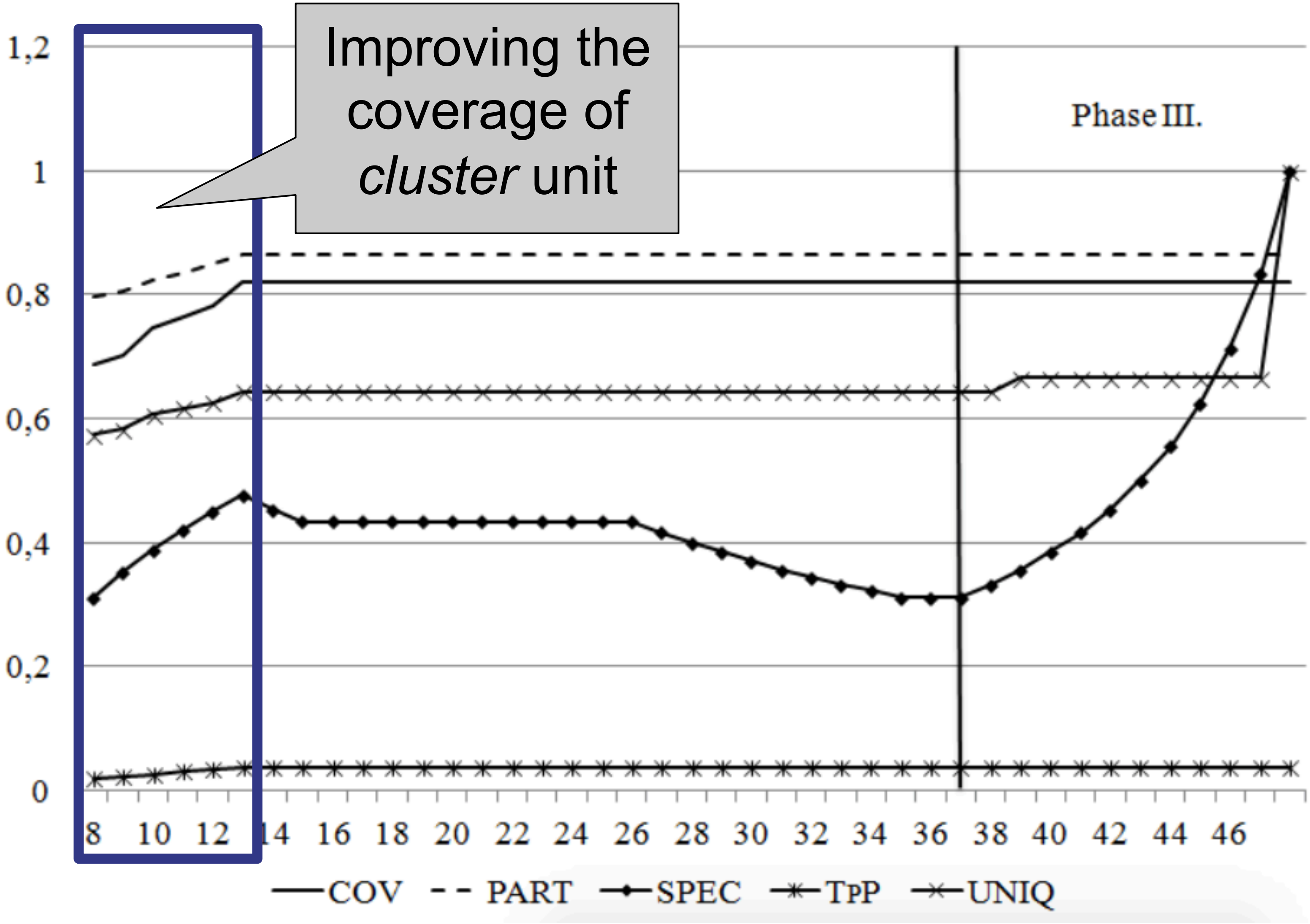


Func. unit	Tests (before)	Tests (after)	Procedures	Statements
cluster	1	10	36	263
data	86	89	213	1588
fl-technique	2	4	16	175
io	13	16	56	429
metric	3	18	60	549
prioritization	2	6	21	159
reader	4	13	35	431
reduction	0	8	33	414
other	1	1	145	331
SoDA	112	165	615	4339

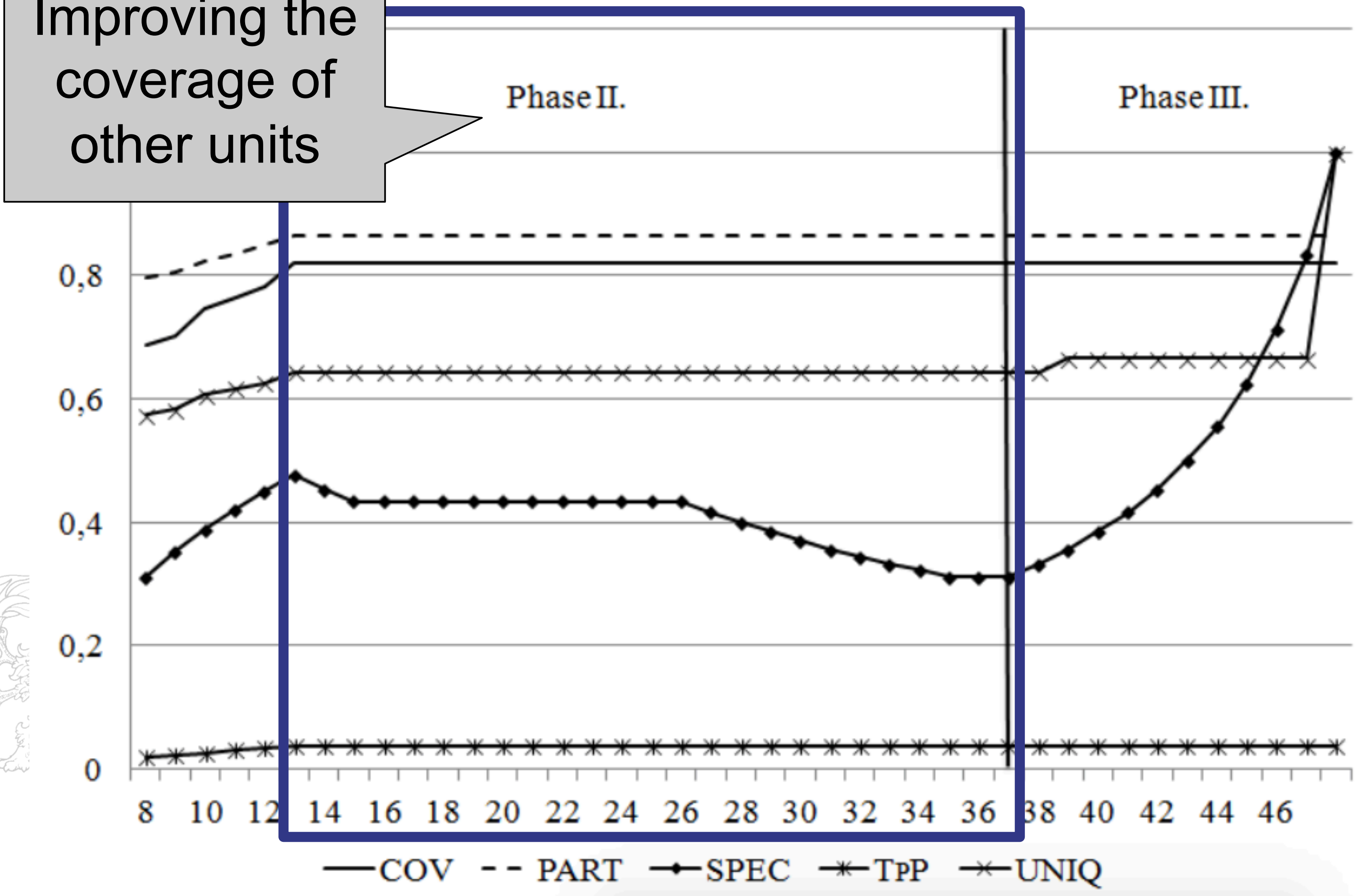
Improvement of SoDA



Improvement of the *cluster* unit


Improving the coverage of *cluster* unit

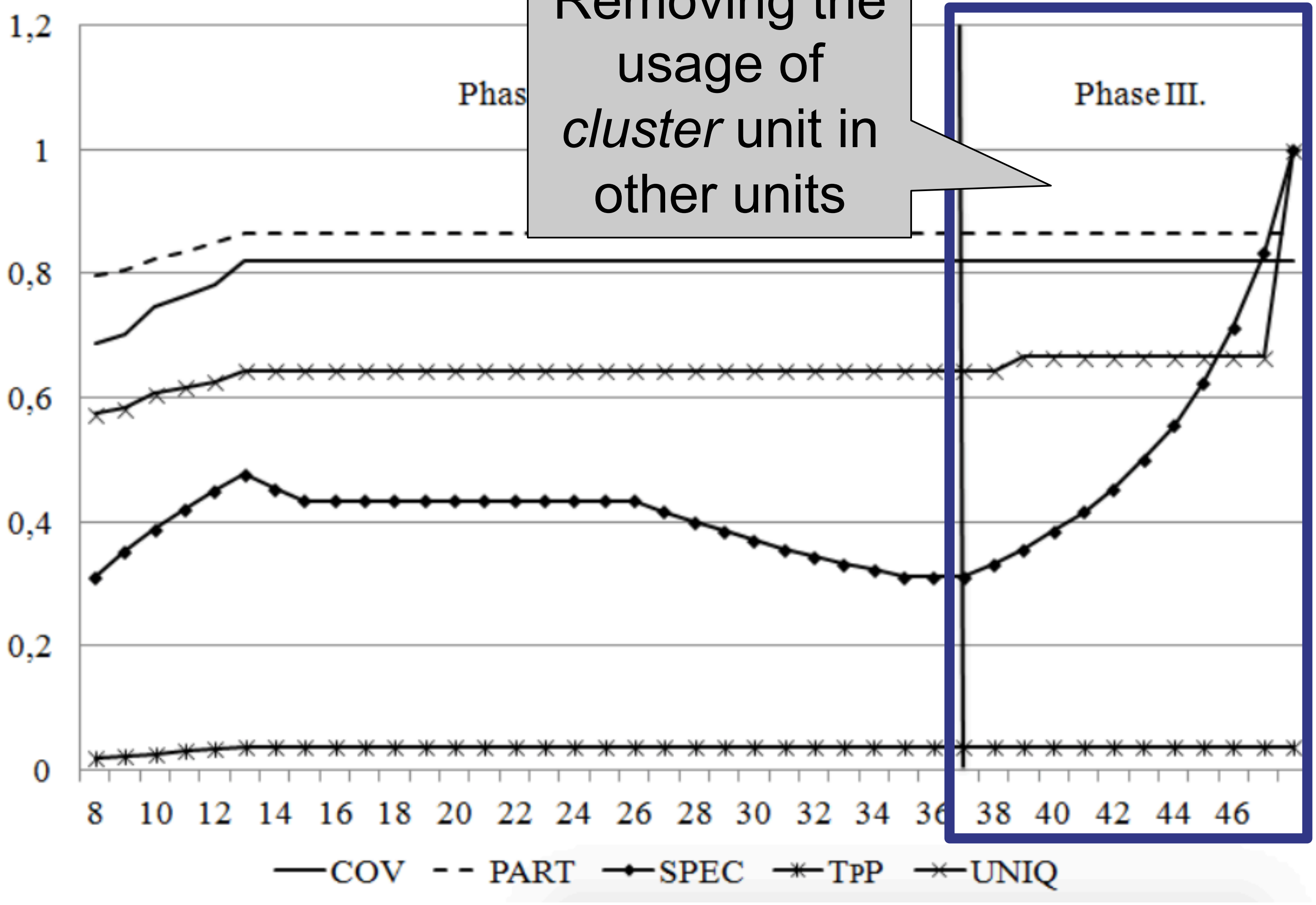
Improvement of the *cluster* unit



Improving the coverage of other units

UNIVERSITAS SCIENTIARUM SZEGEDIENSIS
UNIVERSITY OF SZEGED
Department of Software Engineering

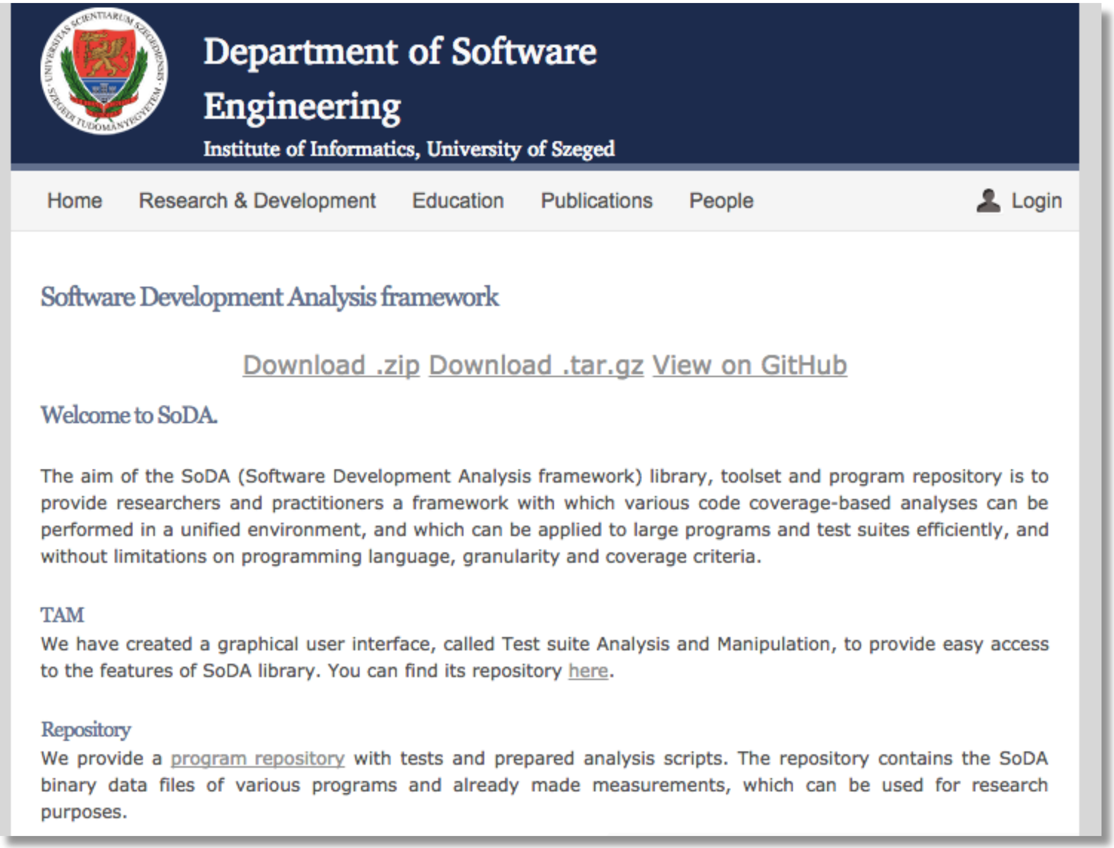
Improvement of the *cluster* unit



Removing the usage of *cluster* unit in other units

TAIME

- One-shot refactoring
- White-box test design
- Change-oriented test suite evolution
- Assessment


Department of Software Engineering

Institute of Informatics, University of Szeged

Software Development Analysis framework

Download .zip Download .tar.gz View on GitHub

Welcome to SoDA.

The aim of the SoDA (Software Development Analysis framework) library, toolset and program repository is to provide researchers and practitioners a framework with which various code coverage-based analyses can be performed in a unified environment, and which can be applied to large programs and test suites efficiently, and without limitations on programming language, granularity and coverage criteria.

TAM

We have created a graphical user interface, called Test suite Analysis and Manipulation, to provide easy access to the features of SoDA library. You can find its repository here.

Repository

We provide a program repository with tests and prepared analysis scripts. The repository contains the SoDA binary data files of various programs and already made measurements, which can be used for research purposes.

http://soda.sed.hu