Improving Continuous Integration with Similarity-based Test Case Selection


Francisco Gomes de Oliveira Neto
gomesf@chalmers.se

Azeem Ahmad
azeem.ahmad@liu.se

Kristian Sandahl
kristian.sandahl@liu.se

Ola Leifler
ola.leifler@liu.se

Eduard Enoiu
eduard.paul.enoiu@mdh.se

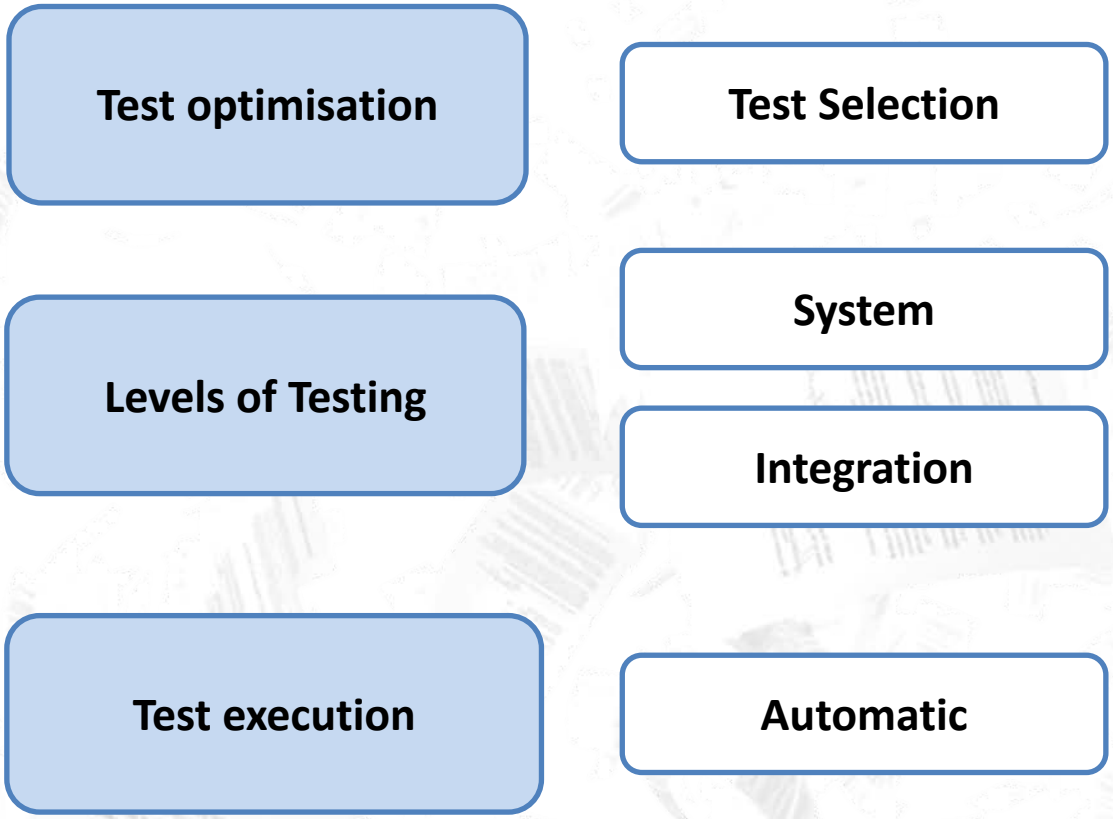
AST 2018
Gothenburg, Sweden

Software Center

Software Center

The problem

The big picture



The big picture

How do we **select them**?

Test platform

Services

Automated system/integration tests

1000+ test cases

3+ hours

Do we **need** that many?

How can we **reduce feedback cycles**?

In a nutshell…

TC1 - New game (new hero)
Select “New game”
Create a new hero
Save hero
Start game

TC2 - New game (random hero)
Select “New game”
Generate a random hero
Save hero
Start game

TC3 – Load game (random hero)
Select “Load game”
Select a previous game file
Load the game

Which tests **would you choose**?

1. TC1 and TC2
2. **TC1 and TC3**
3. **TC2 and TC3**

Proposed solution

- Use **similarity-based** test case selection (SBTCS)
- Avoid executing similar tests
  - Focus on the **test's content**
- Note: We are not using Adapt. Random Testing

In a nutshell…

TC1 - New game (new hero)	TC2 - New game (random hero)	TC3 – Load game
Select “New game”	Select “New game”	Select “Load game”
Create a new hero	Generate a random hero	Select a previous game file
Save hero	Save hero	Load the game
Start game	Start game	

Related Work - SBTCS

- Different **"types"** of similarity:
  - Text: [Cartaxo et al., 2007], [Ledru et al., 2011]
  - Failure history: [Noor and Hemmati, 2015]
  - Modifications: [de Oliveira Neto et al., 2016]
  - Models: [Cartaxo et al., 2011], [Hemmati et al., 2013]
  - Requirements: [Zhang et al., 2018]

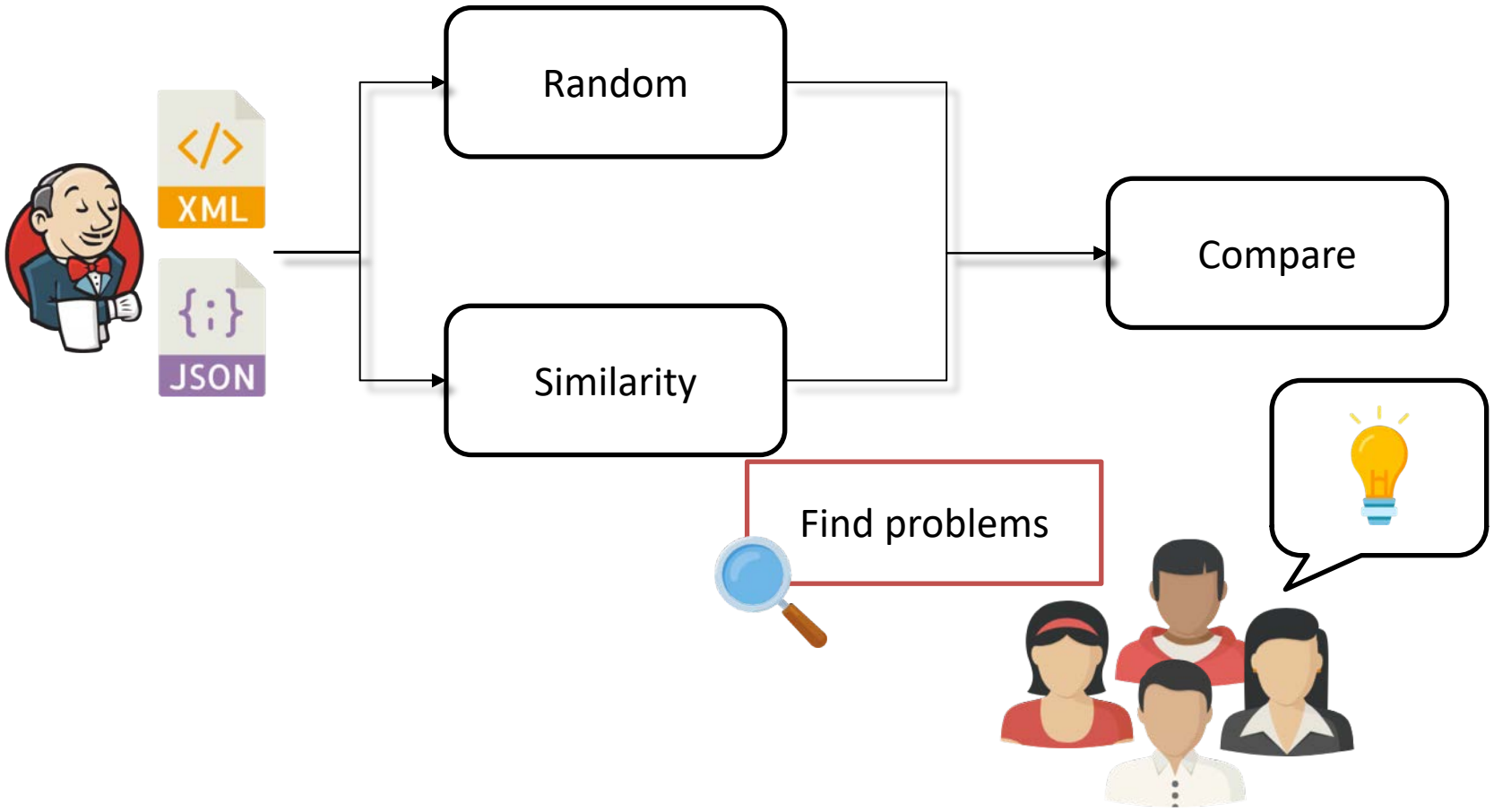
- Rapid releases and prioritization:
  - [Hemmati et al., 2015]

Study with Companies

- **Data** collection: NDA protected

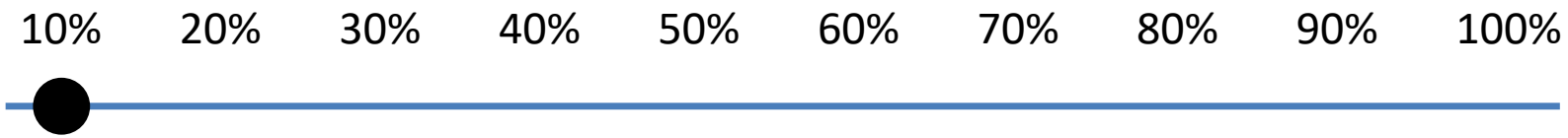
Context	Automated testing in CI pipelines
The cases	Company A: Surveillance company Company B: Automotive company
Analysis:	Comp. A: Coverage & Time Comp. B: Coverage
Data collection:	Archival data and metrics

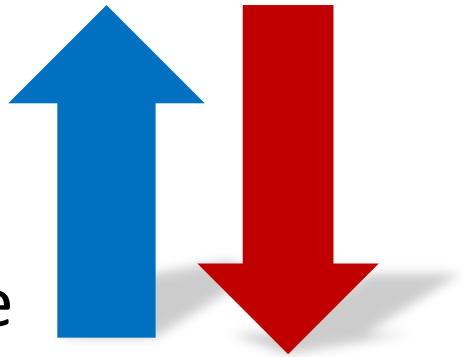
The big picture



How did we do it?

- Explore **thresholds** to remove:



Coverage  Time  Failures??

Let's talk about coverage…

- Coverage of:
  - **Tested** features: Features under test
  - **Required** features: Dependencies to TC execution
  - Test **Steps**: "Standardized" Natural language.

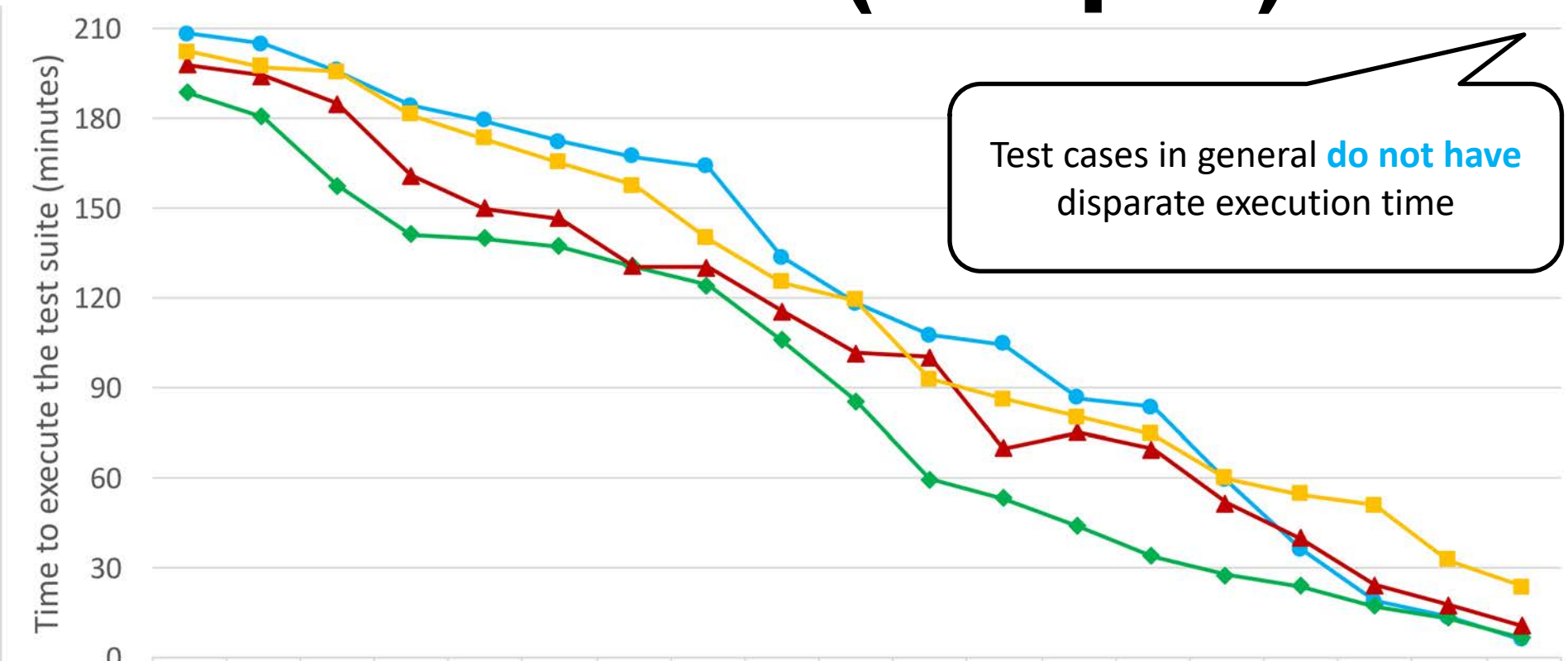
About the data

Company A (mature infrastructure)	Company B (unstable infrastructure)
1000+ test cases	**1500+** test cases
158 tested features	No features
384 combinations of required features	
225 minutes (**3.7 hours**)	

Four different levels:

- **NL** : Normalized Levenshtein Distance
- **JI** : Jaccard Index
- **NCD**: Normalized Compressed Distance
- **RDM**: Random

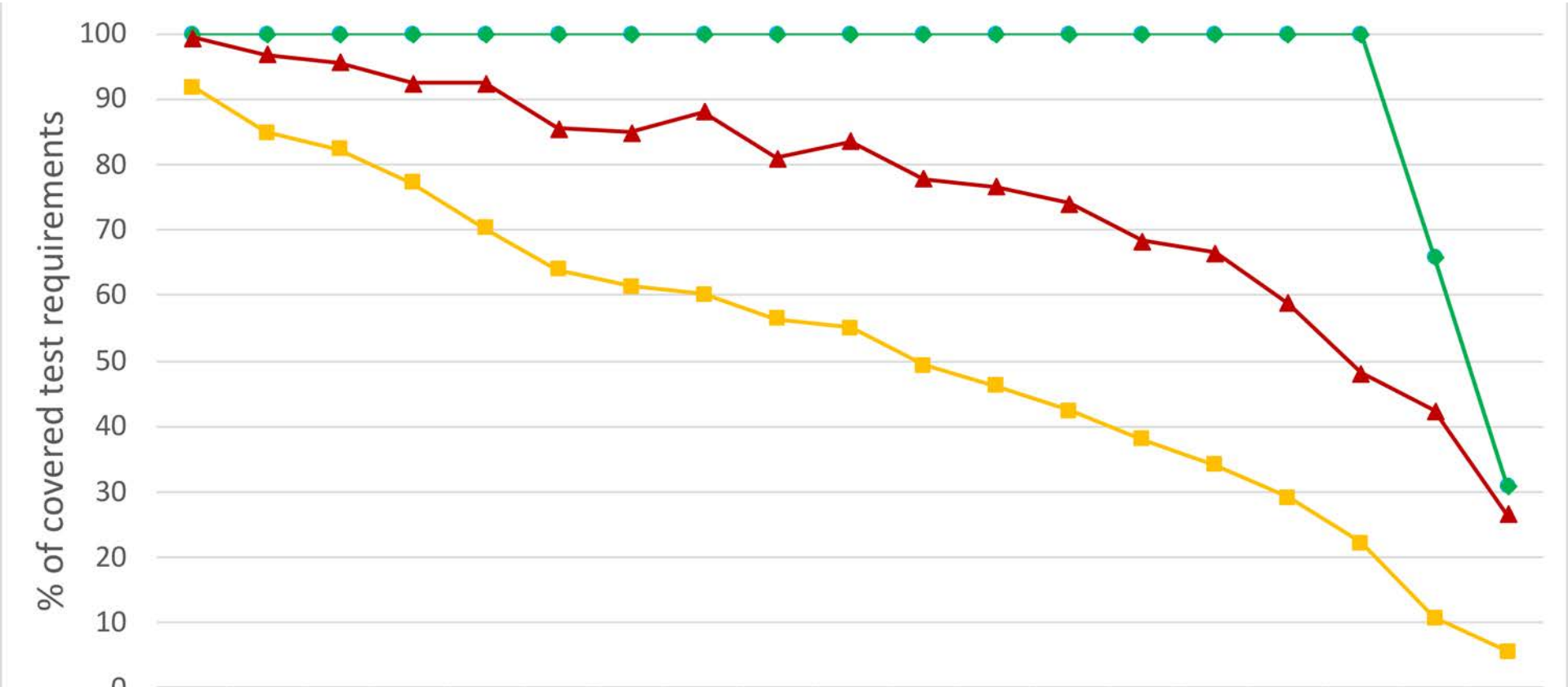
Time Reduction (Comp. A)



	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
JI	208	205	196	185	179	173	168	164	134	118	108	105	87	84	59	36	19	14	6
NL	189	181	158	142	140	138	131	125	106	86	60	54	44	34	28	24	17	13	7
RDM	198	195	185	161	150	147	131	130	116	102	101	70	75	70	52	40	24	18	11
NCD	202	197	196	182	174	165	158	140	125	120	93	87	80	75	60	55	51	33	24

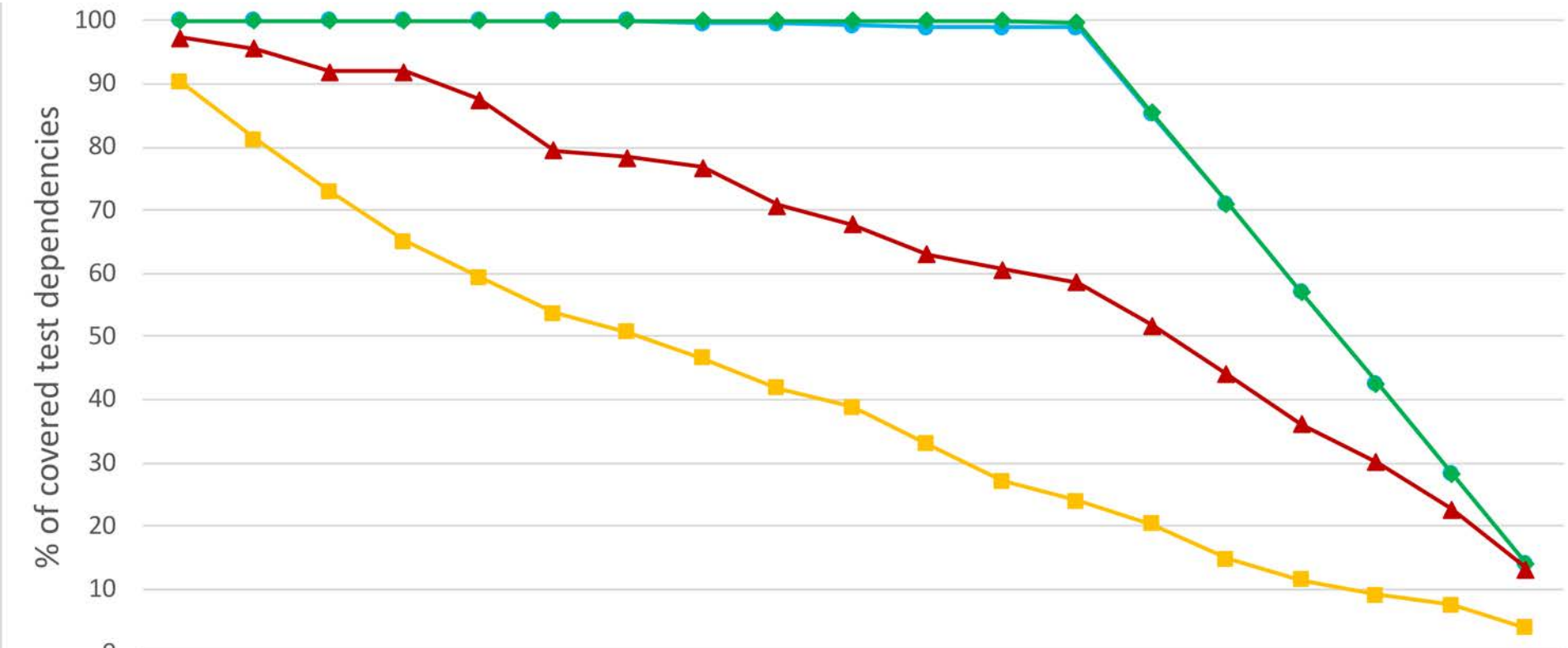
3 hours — 1.5 hours — 6 - 30 minutes

Coverage reduction (Comp. A)



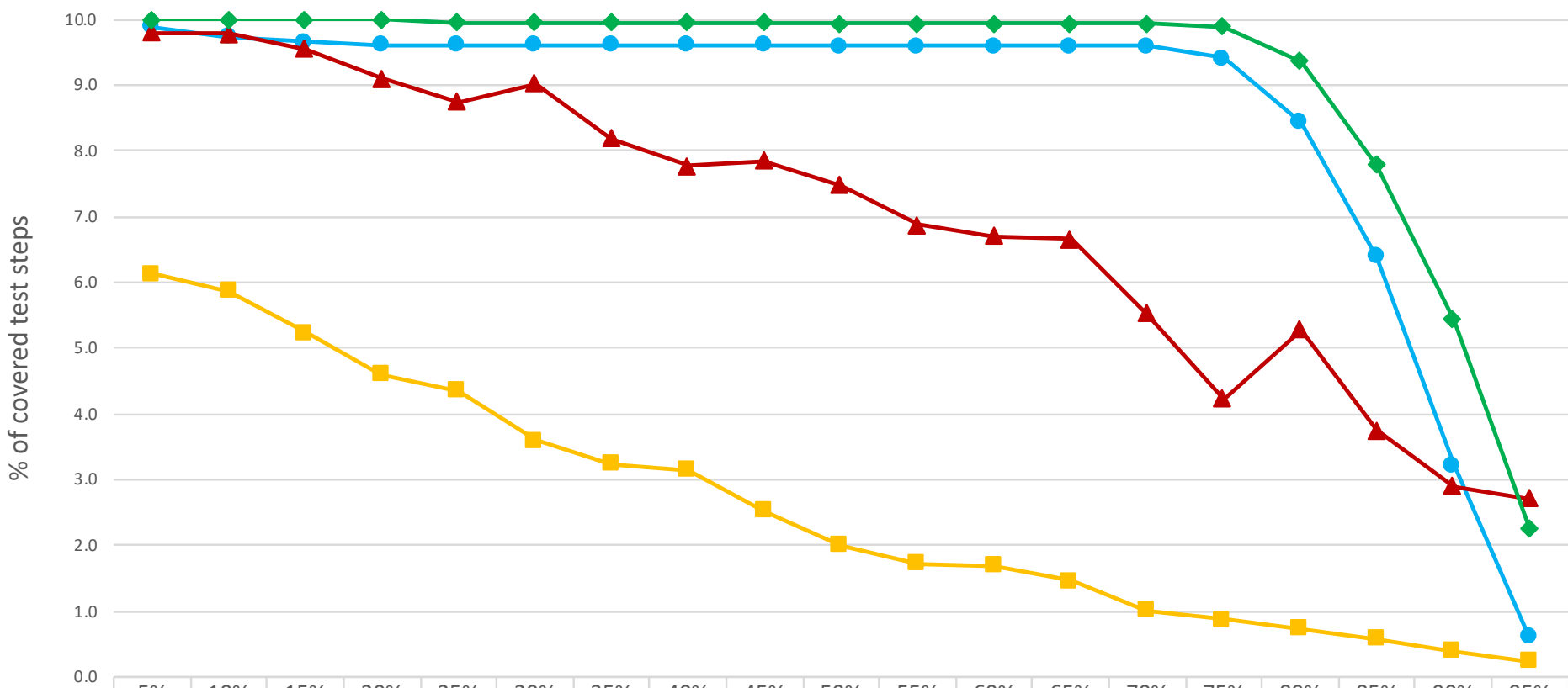
	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
JI	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	66	31
NL	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	66	31
RDM	99	97	96	92	92	85	85	88	81	84	78	77	74	68	66	59	48	42	27
NCD	92	85	82	77	70	64	61	60	56	55	49	46	42	38	34	29	22	11	6

Coverage reduction (Comp. A)



	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
JI	100	100	100	100	100	100	100	99	99	99	99	99	99	85	71	57	43	28	14
NL	100	100	100	100	100	100	100	100	100	100	100	100	100	85	71	57	43	28	14
RDM	97	96	92	92	88	79	78	77	71	68	63	61	59	52	44	36	30	23	13
NCD	90	81	73	65	59	54	51	47	42	39	33	27	24	20	15	11	9	8	4

Coverage reduction (Comp. B)



	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%
JI	9.9	9.7	9.7	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.4	8.4	6.4	3.2	0.6
RDM	9.8	9.8	9.5	9.1	8.7	9.0	8.2	7.8	7.8	7.5	6.9	6.7	6.6	5.5	4.2	5.3	3.7	2.9	2.7
NL	10.0	10.0	10.0	10.0	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.4	7.8	5.4	2.3
NCD	6.1	5.9	5.2	4.6	4.3	3.6	3.2	3.1	2.5	2.0	1.7	1.7	1.5	1.0	0.9	0.7	0.6	0.4	0.2

Some findings

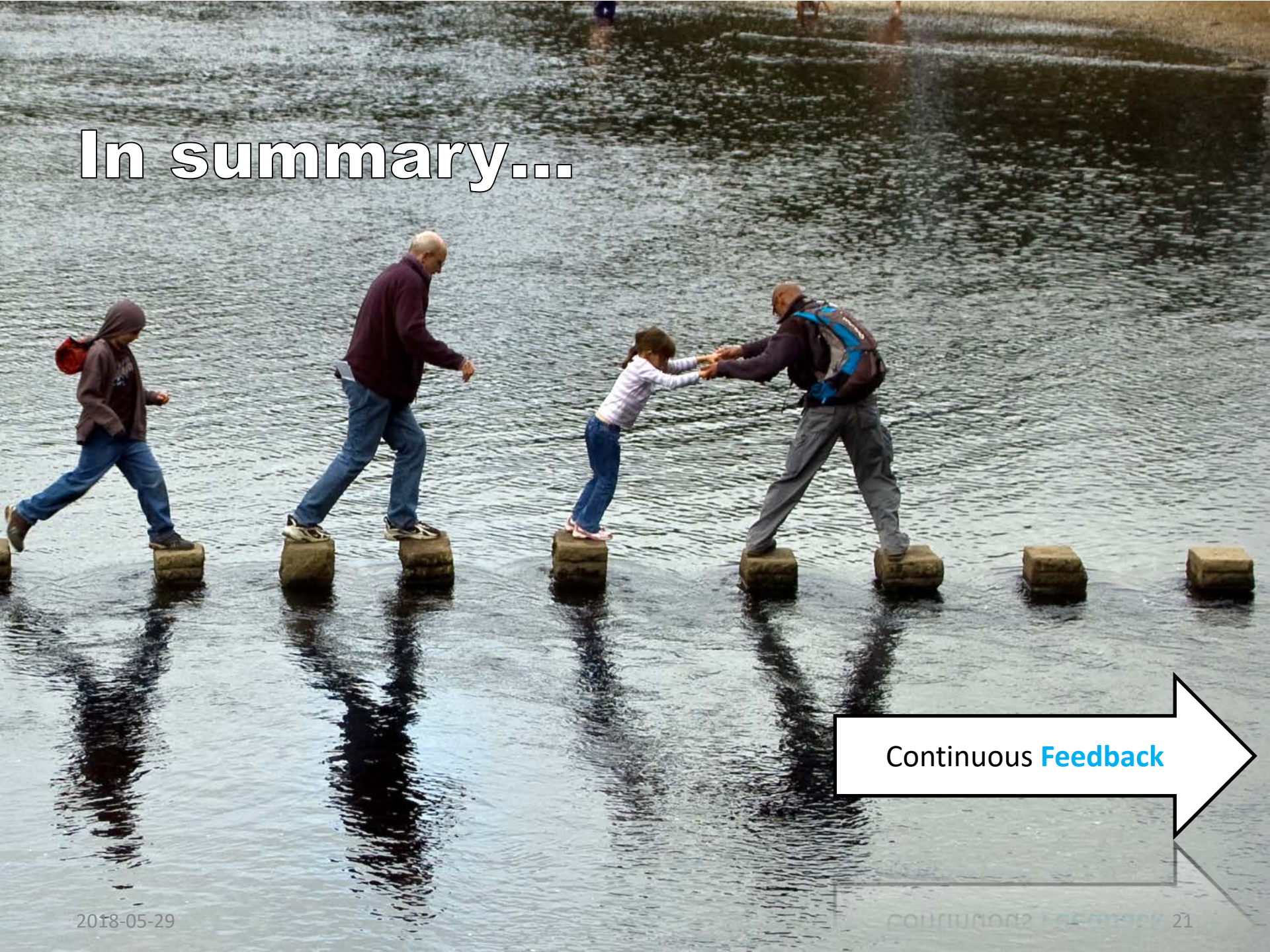
- **Lots** of repetition: Not a bad thing!
- **Unaware** that there were too much repetition!
  - Lack of appropriate maintenance of test cases
  - No one is watching
  - **Copy and paste** of test cases
- Becomes **"wasted information"** on CI cycles

Some findings

- Some **good options**: 50% reduction
  - Consistently safe at coverage
  - 100% coverage of required and tested
  - ca 2 hours fasters.
  - Techniques took less than 1 second to execute.

- **Similarity** and test maintenance
  - Feedback on test **redundancy**
  - **Confirms** existing findings

In summary…



Continuous **Feedback**

References

Y. Ledru, A. Petrenko, S. Boroday, and N. Mandran, “Prioritizing test cases with string distances,” ASE 2011

E. G. Cartaxo, F. G. de Oliveira Neto, and P. D. L. Machado. Automated Test Case Selection Based on a Similarity Function. MOTES 2007

E. G. Cartaxo, P. D. L. Machado, and F. G. de Oliveira Neto. On the use of a similarity function for test case selection in the context of model-based testing. STVR. 2011

F. G. de Oliveira Neto, R. Torkar, and P. D. L. Machado. Full modification coverage through automatic similarity-based test case selection. IST 2016

H. Hemmati, A. Arcuri, and L. Briand Achieving Scalable Model-based Testing Through Test Case Diversity. TOSEM 2013

H. Hemmati, Z. Fang, and M. V. Mantyla. Prioritizing Manual Test Cases in Traditional and Rapid Release Environments. ICST 2015

T. B. Noor and H. Hemmati. A similarity-based approach for test case prioritization using historical failure data. ISSRE 2015

Questions??


Francisco Gomes de Oliveira Neto
gomesf@chalmers.se

Azeem Ahmad
azeem.ahmad@liu.se

Kristian Sandahl
kristian.sandahl@liu.se

Ola Leifler
ola.leifler@liu.se

Eduard Enoiu
eduard.paul.enoiu@mdh.se

AST 2018
Gothenburg, Sweden

Software Center