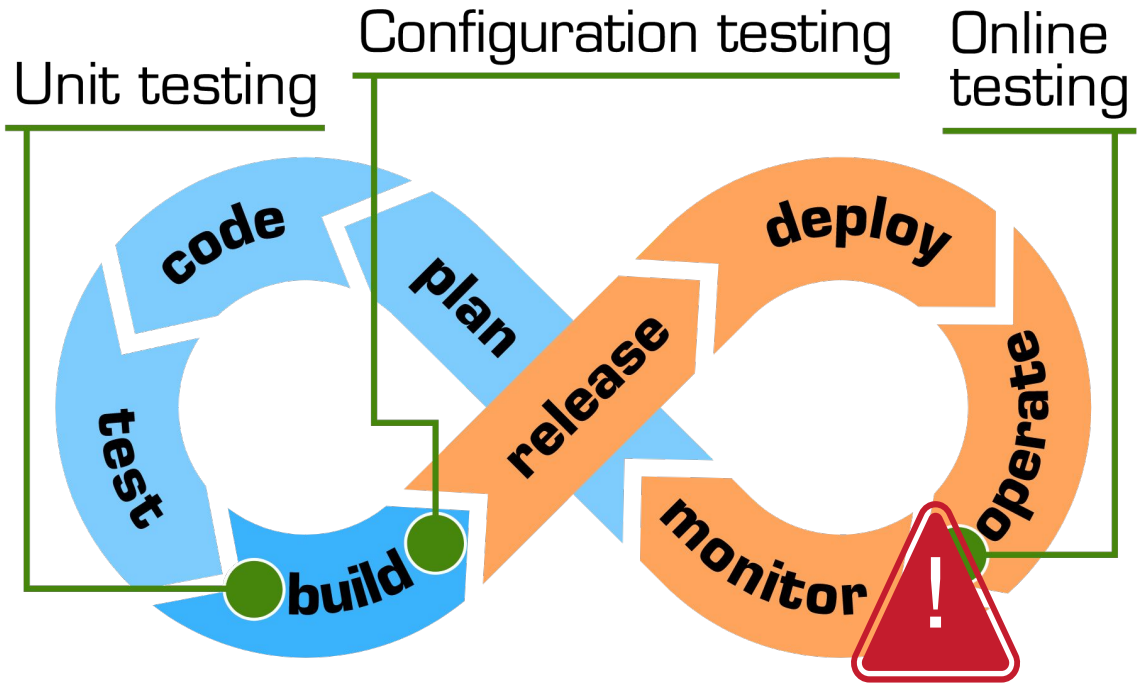
Crash reproduction difficulty, an initial assessment

Boris Cherry
Xavier Devroey
Pouria Derakhshanfar
Benoît Vanderose

Crash Reproduction




```
java.lang.ClassCastException: […]
  at org…..SolrEntityReferenceResolver.getWikiReference(....java:93)
  at org…..SolrEntityReferenceResolver.getEntityReference(….java:70)
  at org…..SolrEntityReferenceResolver.resolve(….java:63)
  at org…..SolrDocumentReferenceResolver.resolve(….java:48)
  at …
```

Crash Reproduction

Write a test case reproducing the crash

Describe the bug

Eclipse Save Action Crash - `@Getter` + `@Setter` or `@Data` in combination with action either "Use "this" qualifier for field accesses" or "Use "this" qualifier for method accesses". Eclipse throws an error when the "Use "this" qualifier for field accesses" is configured to "Only if necessary". In other words, only if they are both set to "Always" will no error be thrown.

```
A save participant caused problems.
  The save participant 'Code Clean Up' caused an exception: org.eclipse.text.edits.MalformedTreeException: Overl
```

To Reproduce

1. Install latest Eclipse version (does not appear to be Eclipse version related but tested in 2020-03 and 2020-09)
2. Install lombok 1.18.14 for aforementioned eclipse version
3. Preferences -> Java -> Editor -> Save Actions -> Enable "Perform the selected actions on save" -> Enable "Additional actions"
4. "Configure..." -> Member Accesses -> Enable "Use "this" qualifer for field (or member, or both) accesses -> Select "Only if necessary"
5. Add POJO annotated with either `@Getter` and `@Setter`, or just `@Data` with at least 2 fields (no error with just 1 field)
6. Attempt to save file

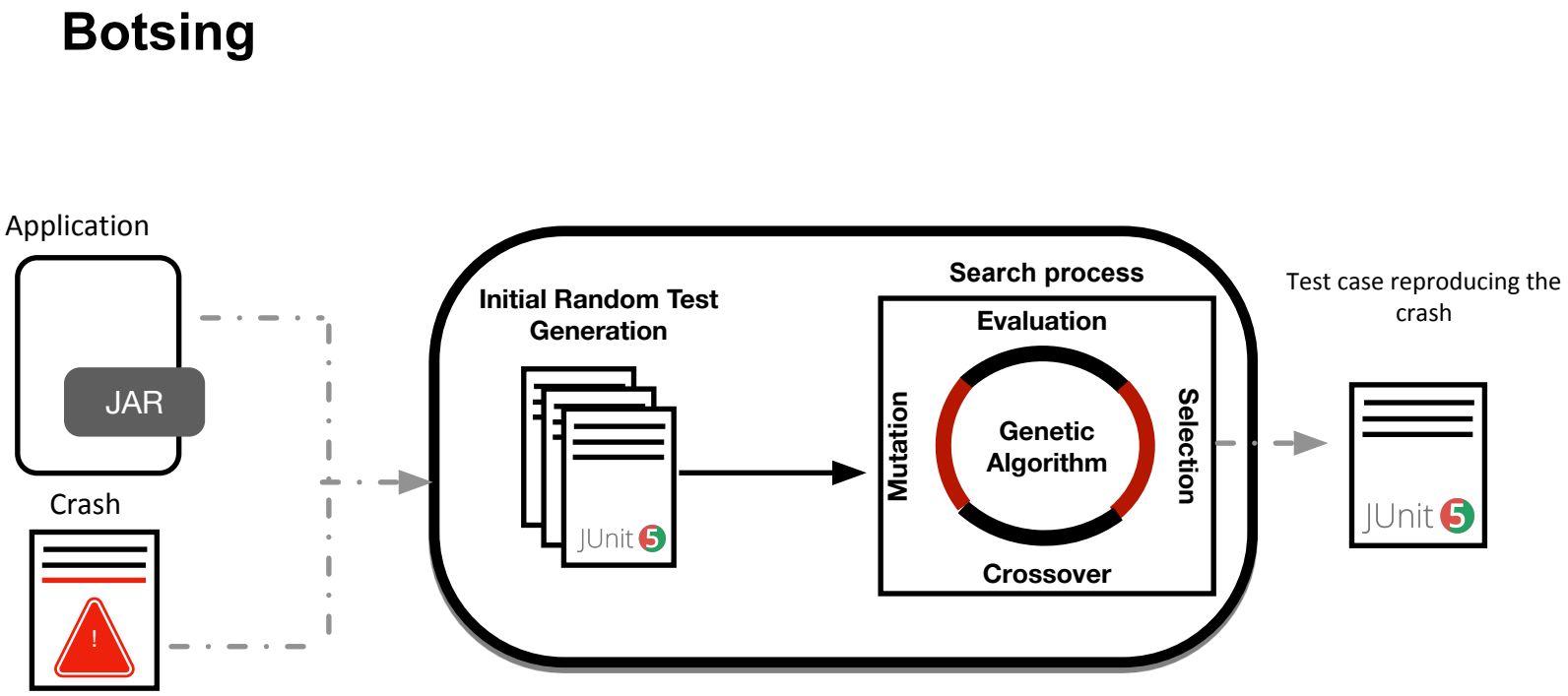
Manually :

- Can be laborious
- Requires knowledge

Automated crash reproduction

- Symbolic execution
- Model checking
- …
- → **Search-based test generation**

Search-based crash reproduction



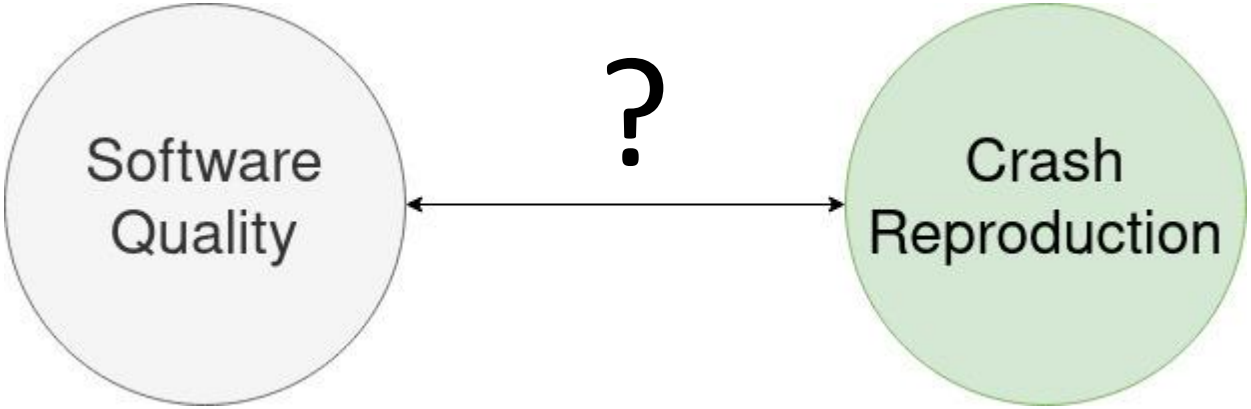
Goals

- Characterize difficulty for search-based crash reproduction
- Search-based unit test generation uses code size + cyclomatic complexity
- What about crash reproduction?

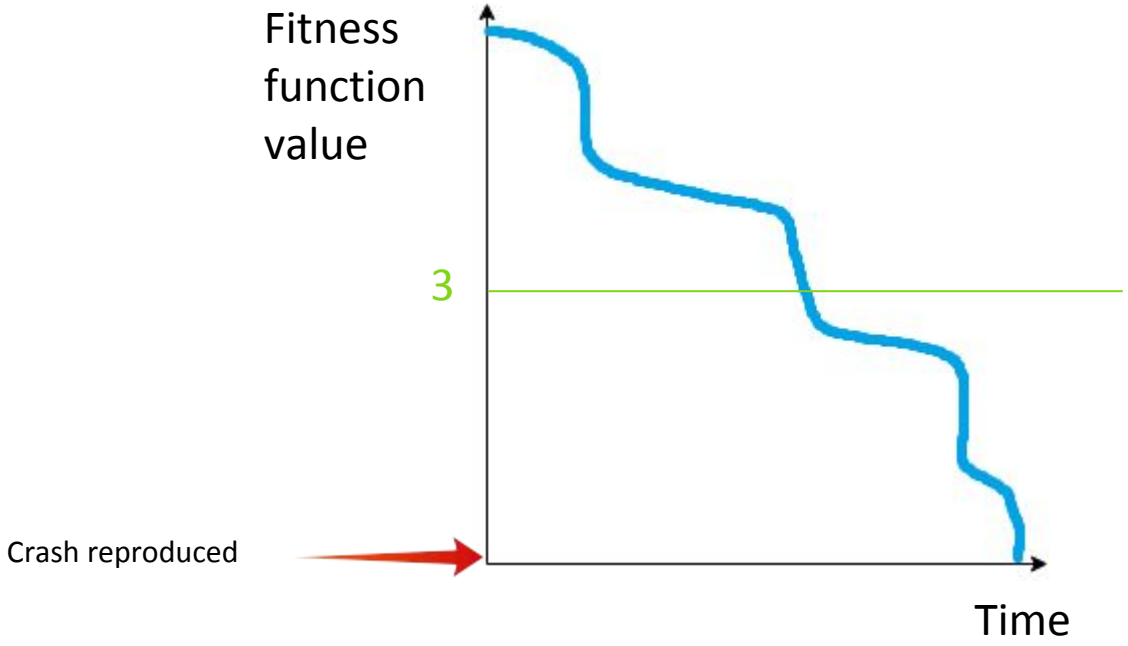
Research question

- ***How do measurable properties of software artefacts influence the difficulty to reproduce a crash using search-based crash reproduction?***

Contribution

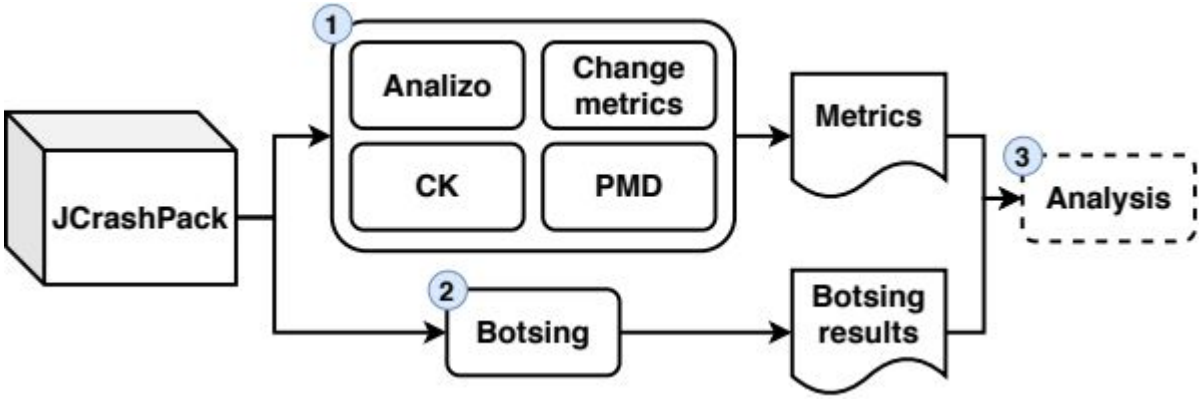


Fitness function



Data gathering

- Each stack trace frame → metrics on class/method of the frame

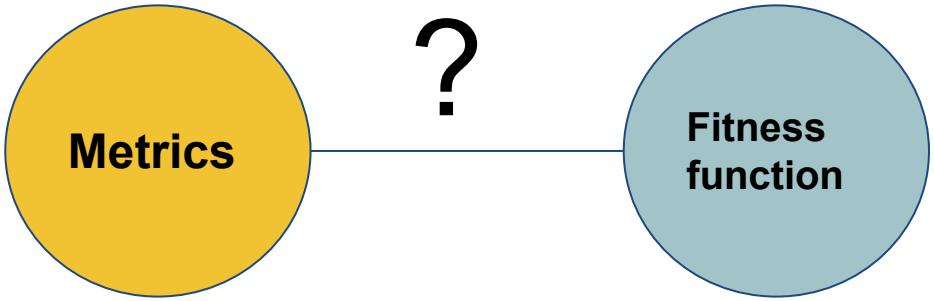


Selected tools

- FOSS tools
- Static code metrics
  - CK
  - Analizo
- Change metrics
  - Custom tool
- Code smells
  - PMD
- Cyclomatic Complexity (McCabe's definition)
  - PMD

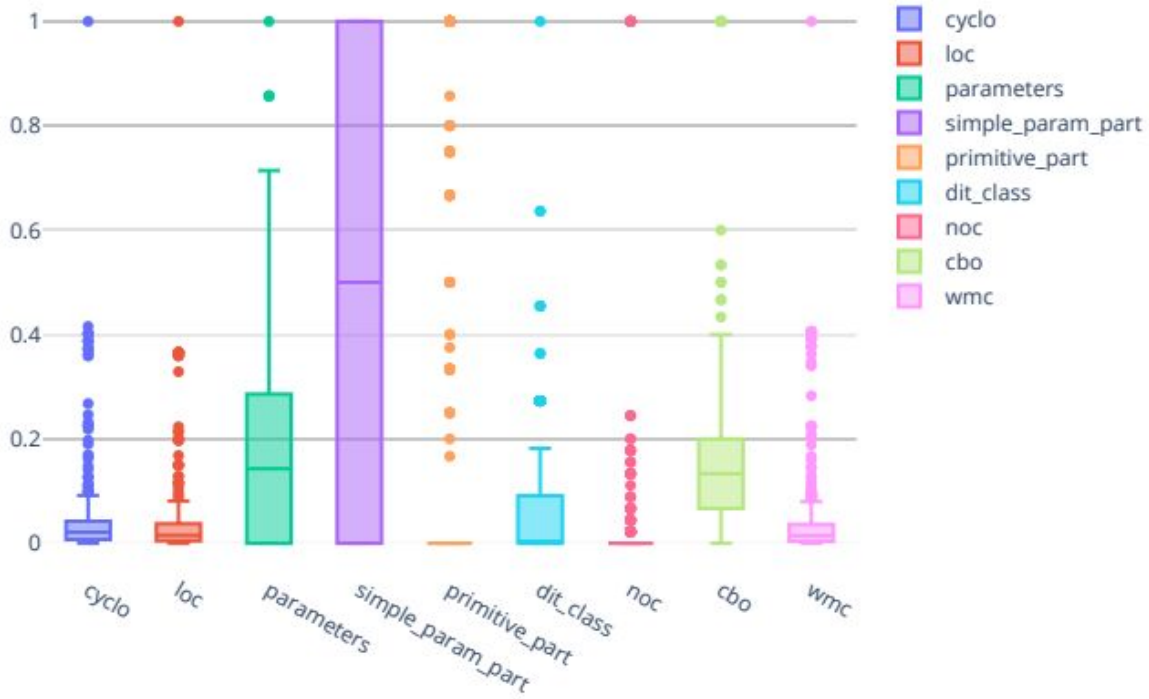
Data Analysis

- Visualisation
- Correlation analysis



Visualisation

Benchmark
Diversity



Correlation

Name	*fs*	ρ	Description
maxNestedBlocks (c)	> 3	0.34	Max. number of nested blocks.
variables (m)	> 3	0.42	Num. of variable declarations.
tryCachQty (m)	> 3	0.31	Num. of *try/catch*.
wmc (c)	> 3	0.34	Class complexity.
loc (m)	> 3	0.44	Num. of line of codes.
cbo (m)	> 3	0.33	Coupling between objects.
maxNestedBlocks (m)	> 3	0.39	Max. number of nested blocks.
rfc (m)	> 3	0.45	Response for a class.
cyclo (m)	> 3	0.37	Cyclomatic complexity.
mmloc (c)	> 3	0.45	Max. method LOC.
simple_param_part (m)	≤ 3	-0.42	Perc. of simple parameters.
primitive_part (m)	≤ 3	-0.40	Perc. of primitive parameters.
cyclo_add (m)	≤ 3	0.54	Lower frames add. complexity.

Data collection

- Few maintained FOSS static analyzers
- Costly operations
  - XWIKI, 106 MB of source code
  - 16 – 21h / version (while powerful server)

Future works

- Include code smells & change metrics in the analysis
- Machine learning
- Search process configuration

Crash reproduction difficulty, an initial assessment

Boris Cherry
Xavier Devroey
Pouria Derakhshanfar
Benoît Vanderose