Overview of SBSE

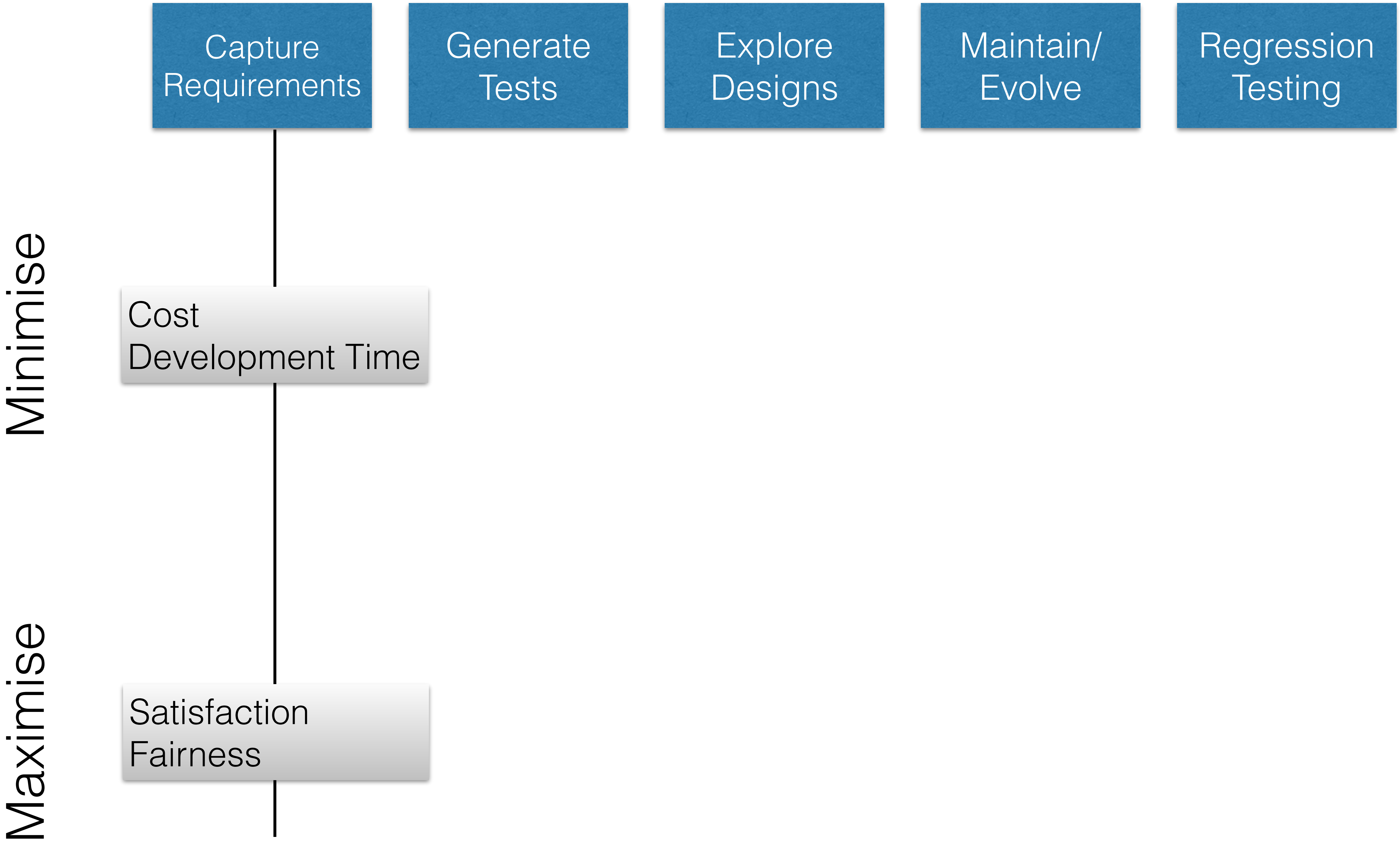
SEP592, Summer 2020
Shin Yoo

Search-Based Software Engineering

- Application of all the optimisation techniques we have seen so far, to the various problems in software engineering.

- Not web search engines :(

- Not code search :(



Capture Requirements

Generate Tests

Explore Designs

Maintain/ Evolve

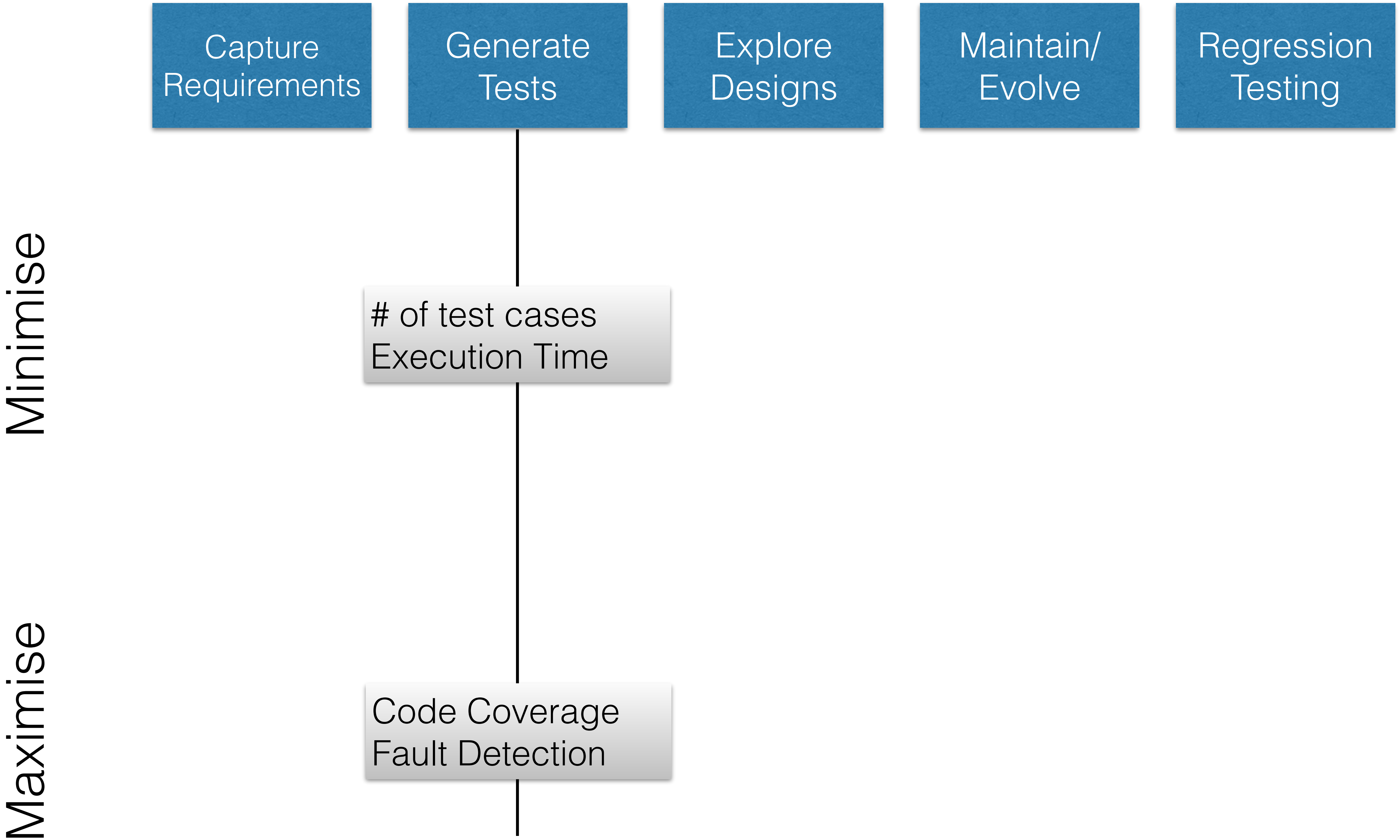
Regression Testing

Minimise

Cost
Development Time

Maximise

Satisfaction
Fairness



Capture Requirements

Generate Tests

Explore Designs

Maintain/ Evolve

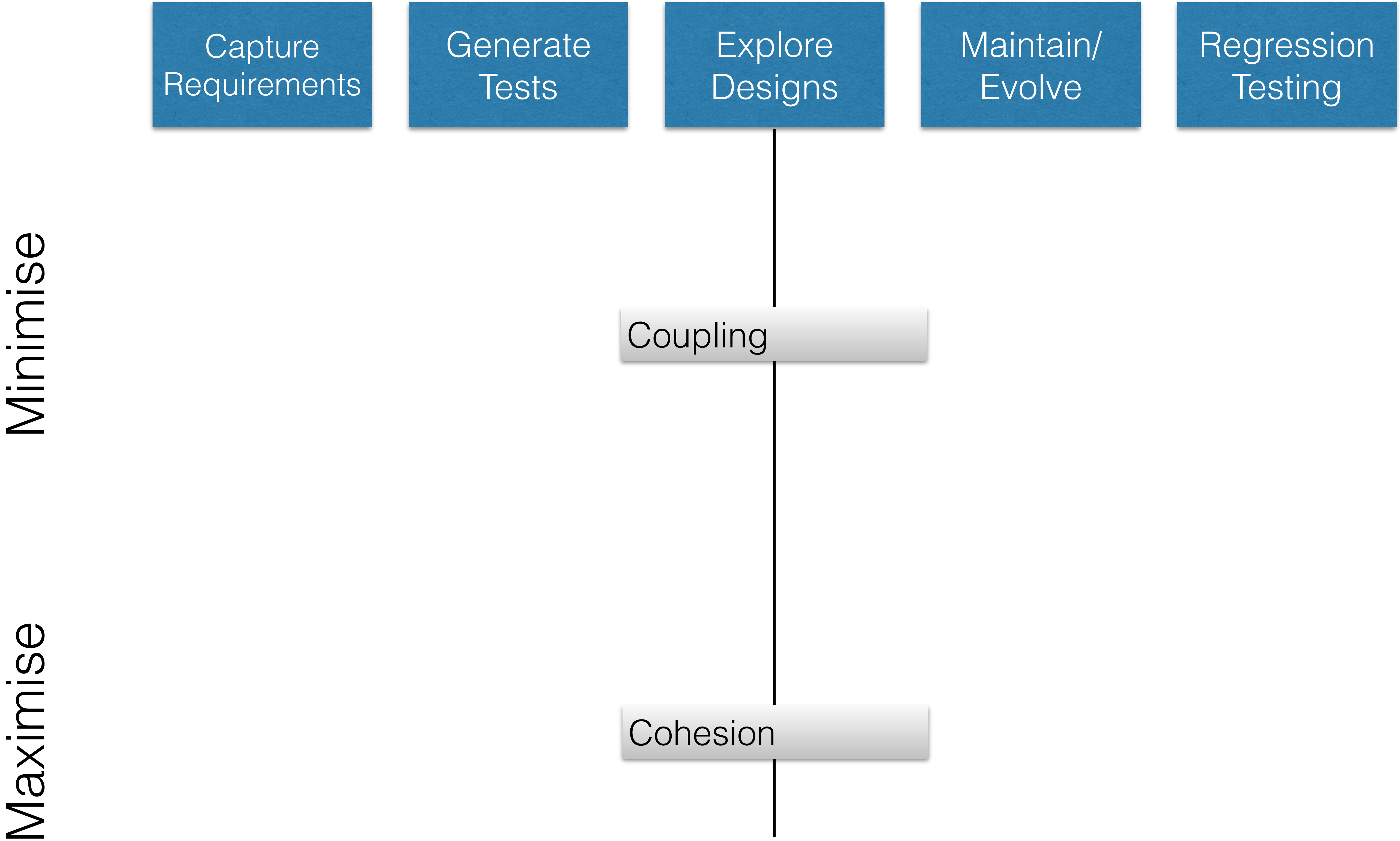
Regression Testing

Minimise

# of test cases
Execution Time

Maximise

Code Coverage
Fault Detection



Capture Requirements

Generate Tests

Explore Designs

Maintain/ Evolve

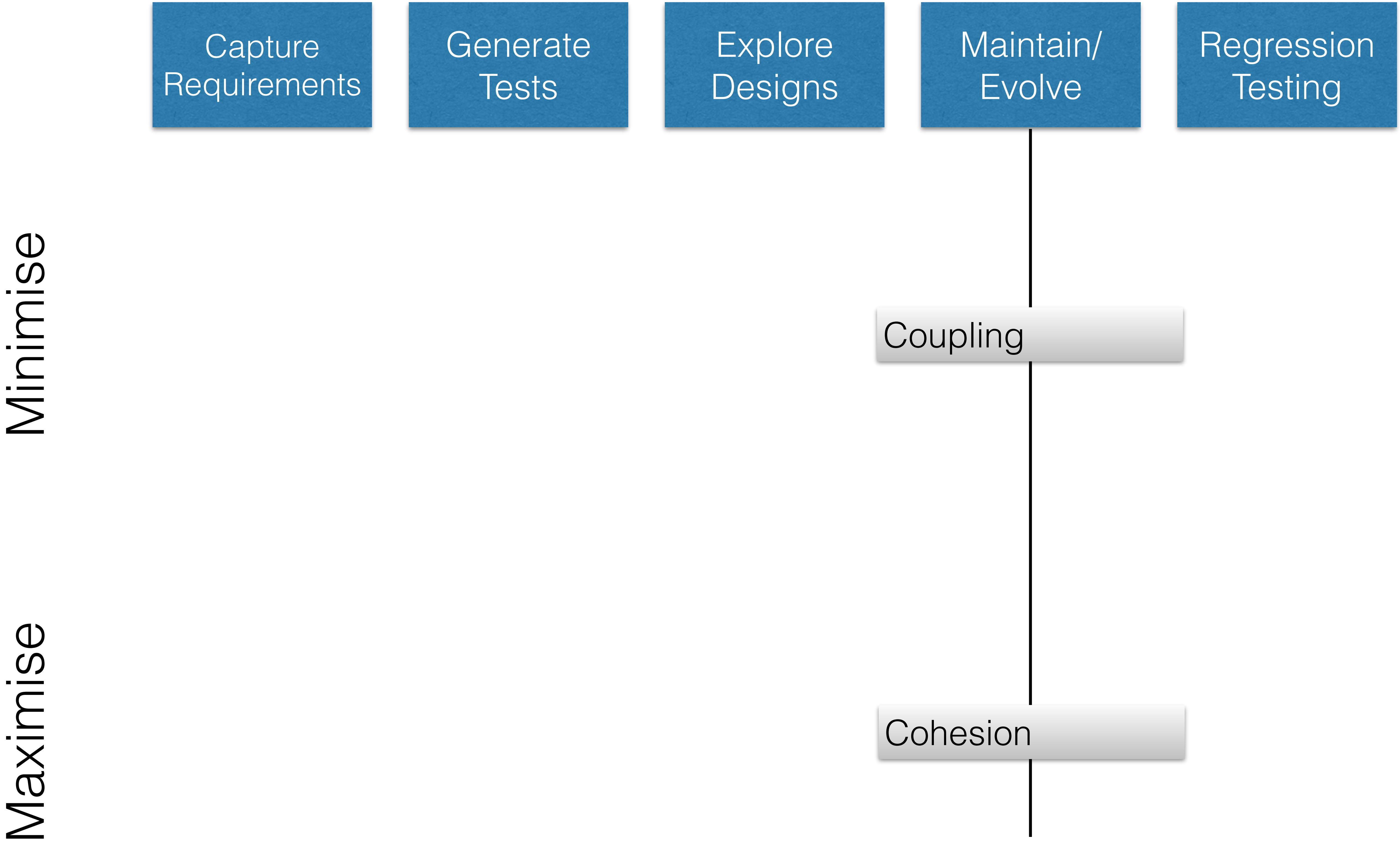
Regression Testing

Minimise

Coupling

Maximise

Cohesion



Capture Requirements

Generate Tests

Explore Designs

Maintain/ Evolve

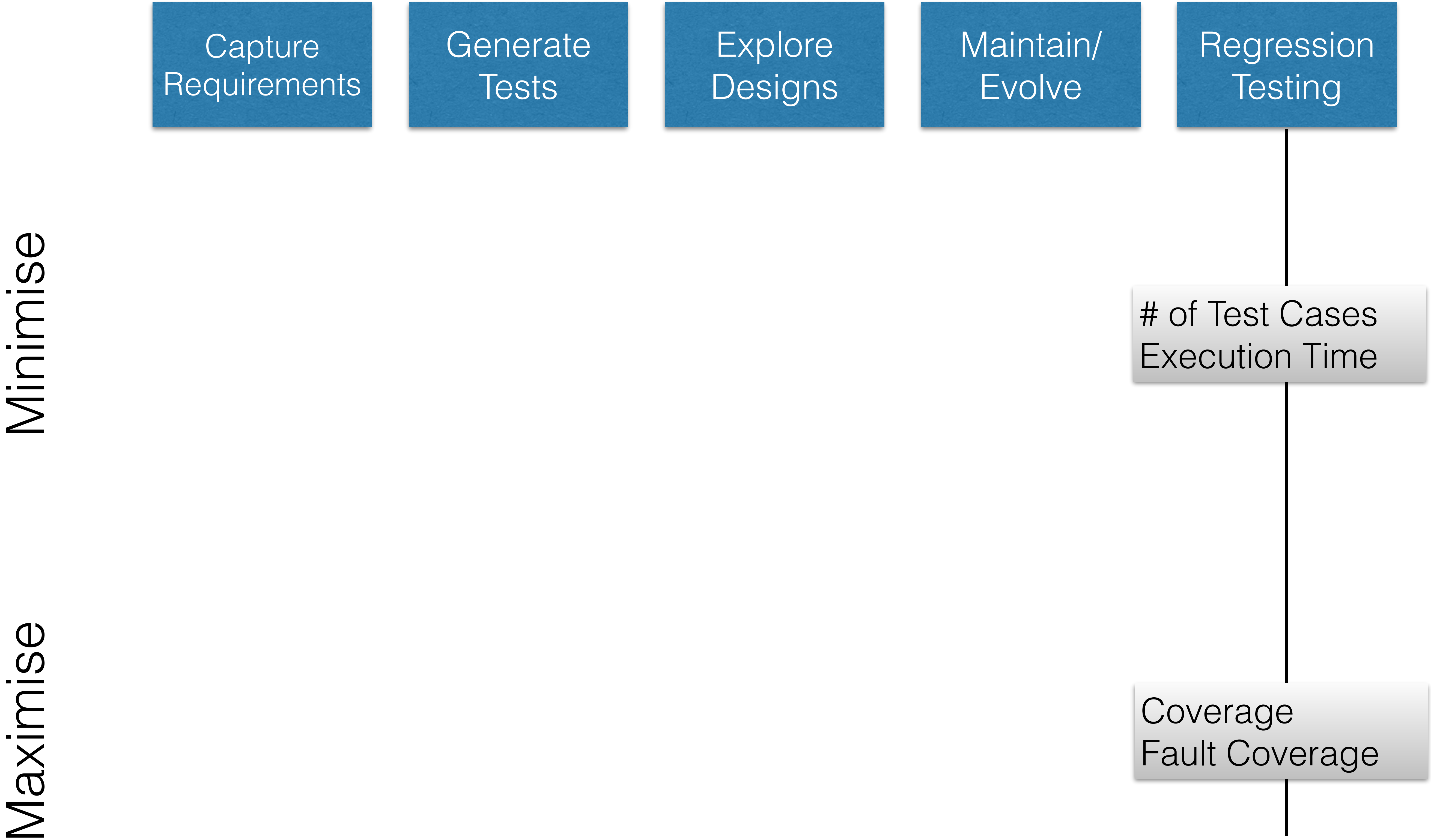
Regression Testing

Minimise

Coupling

Maximise

Cohesion



Capture Requirements

Generate Tests

Explore Designs

Maintain/ Evolve

Regression Testing

Minimise

# of Test Cases
Execution Time

Maximise

Coverage
Fault Coverage

Good Starting Points

- M. Harman. The current state and future of search based software engineering. In FOSE '07: 2007 Future of Software Engineering, pages 342–357,2007.
- M. Harman, S. A. Mansouri, and Y. Zhang. Search-based software engineering: Trends, techniques and applications. ACM Computing Surveys, 45(1):11:1–11:61, December 2012.

Cost Estimation

- Evolve mathematical functions (symbolic regression) that would predict the project development effort based on various input variables.

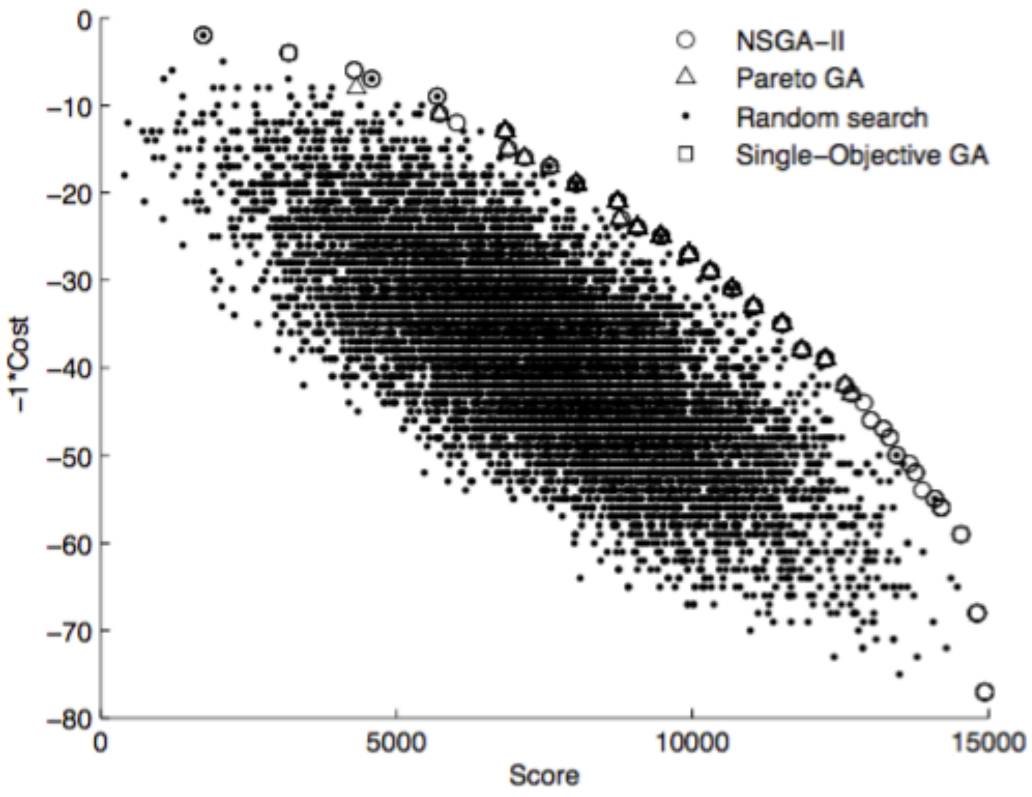
    - J. J. Dolado. A validation of the component-based method for software size estimation. IEEE Transactions on Software Engineering, 26(10):1006–1021, 2000.

Project Planning

- Team allocation to project work packages, including the possibility of abandonment (i.e. work no longer needed/practical) and rework (i.e. additional work needed).
    - G. Antoniol, M. Di Penta, and M. Harman. A Robust Search-based Approach to Project Management in the Presence of Abandonment, Rework, Error and Uncertainty. In Proceedings of the 10th International Symposium on the Software Metrics (METRICS ’04), pages 172–183, Chicago, USA, 11-17 September 2004. IEEE Computer Society.

Next Release Problem

- Find the ideal set of requirements that balances customer requests, resource constraints, and interdependencies between requirements.
  - A. Bagnall, V. Rayward-Smith, and I. Whittley. The next release problem. Information and Software Technology, 43(14):883–890, Dec. 2001.
  - Y. Zhang, M. Harman, and S. A. Mansouri. The Multi-Objective Next Release Problem. In GECCO '07: Proceedings of the 2007 Genetic and Evolutionary Computation Conference, pages 1129–1136. ACM Press, 2007.



(d) 100 customers; 20 requirements

Optimising Source Code

- Random sampling of code transformation to find compiler optimisation
  - K. D. Cooper, P. J. Schielke, and D. Subramanian. Optimizing for reduced code space using genetic algorithms. In Proceedings of the ACM SIGPLAN 1999 Workshop on Languages, Compilers and Tools for Embedded Systems (LCTES‘99), volume 34.7 of ACM SIGPLAN Notices, pages 1–9, NY, May 5 1999. ACM Press.
- Automated Parallelisation
  - K.P.Williams.Evolutionary Algorithms for Automatic Parallelization. PhD thesis, University of Reading, UK, Department of Computer Science, Sept. 1998.

Test Data Generation

- Many, many different approaches and ideas; too many to list all:
    - P. McMinn. Search-based software test data generation: A survey. Software Testing, Verification and Reliability, 14(2):105–156, June 2004.

Regression Testing

- Pareto-efficient Test Suite Minimisation:
  - S. Yoo and M. Harman. Pareto efficient multi-objective test case selection. In Proceedings of International Symposium on Software Testing and Analysis, pages 140–150. ACM Press, July 2007.
- Test Case Prioritisation:
  - Z. Li, M. Harman, and R. M. Hierons. Search Algorithms for Regression Test Case Prioritization. IEEE Transactions on Software Engineering, 33(4):225–237, 2007.
- Multi-objective Prioritisation:
  - M. G. Epitropakis, S. Yoo, M. Harman, and E. K. Burke. Empirical evaluation of pareto efficient multi-objective regression test case prioritisation. In Proceedings of the 2015 International Symposium on Software Testing and Analysis, ISSTA 2015, pages 234–245, New York, NY, USA, 2015. ACM.

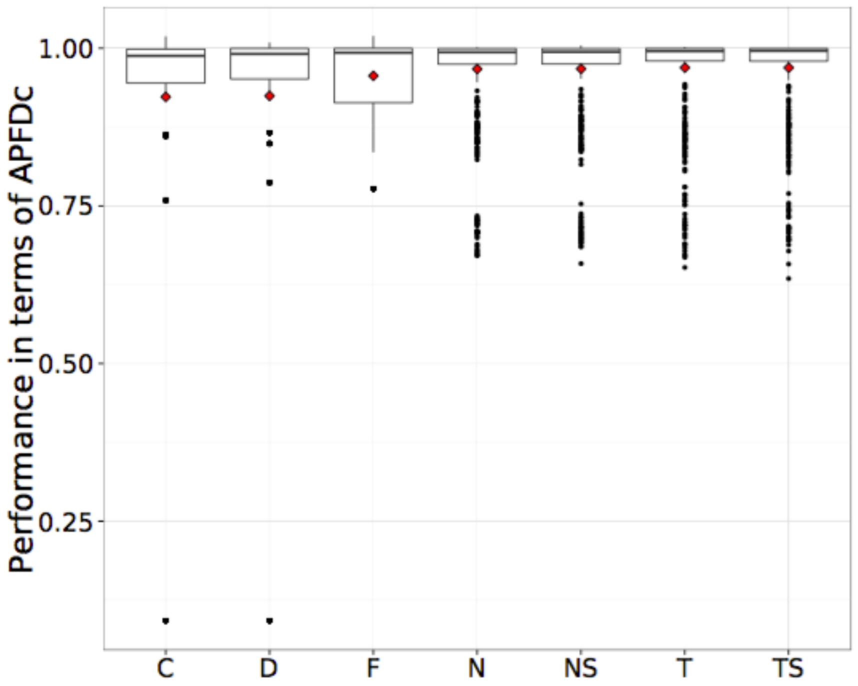


Figure 3: Boxplots of the APFD$_c$ metric across all studied subjects. MOEAs and their variants show higher median values and smaller variances.

Maintenance & Reverse Engineering

- Module Clustering: assign modules to clusters based on their relationships

    - B. S. Mitchell and S. Mancoridis. On the automatic modularization of software systems using the bunch tool. IEEE Transactions on Software Engineering, 32(3):193–208, 2006.

    - K. Praditwong, M. Harman, and X. Yao. Software module clustering as a multi-objective search problem. IEEE Transactions on Software Engineering, 37(2):264–282, March-April 2010.

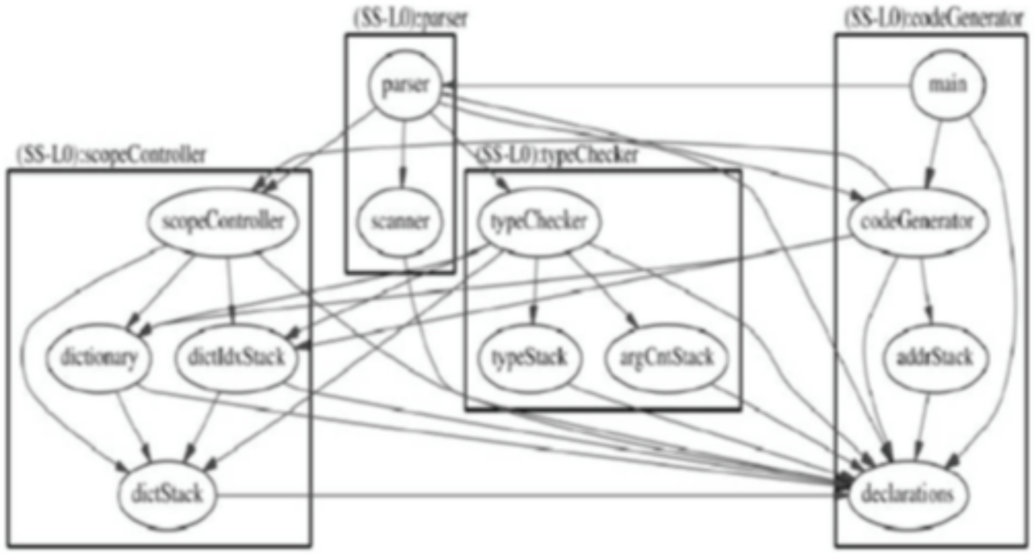
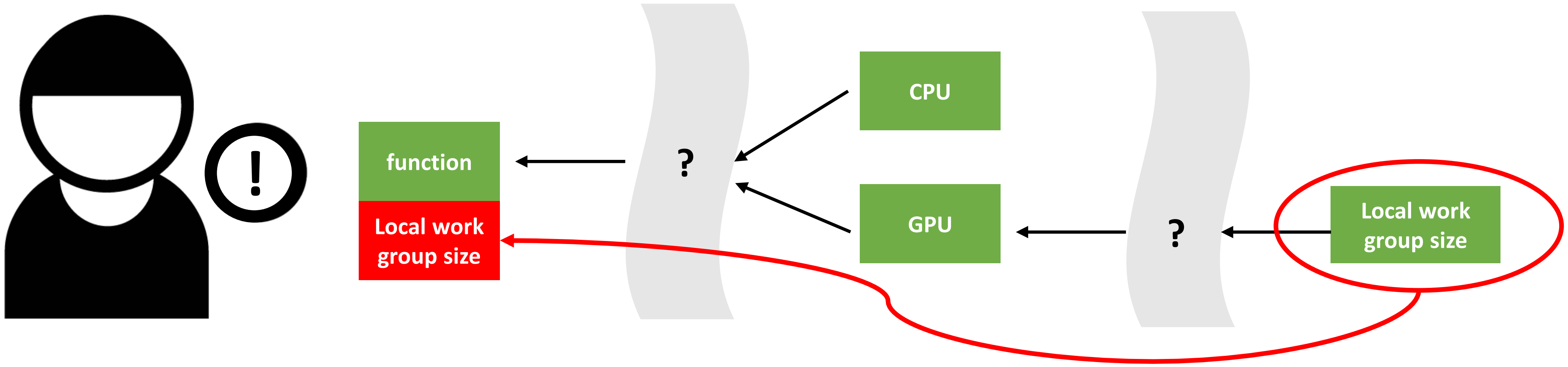


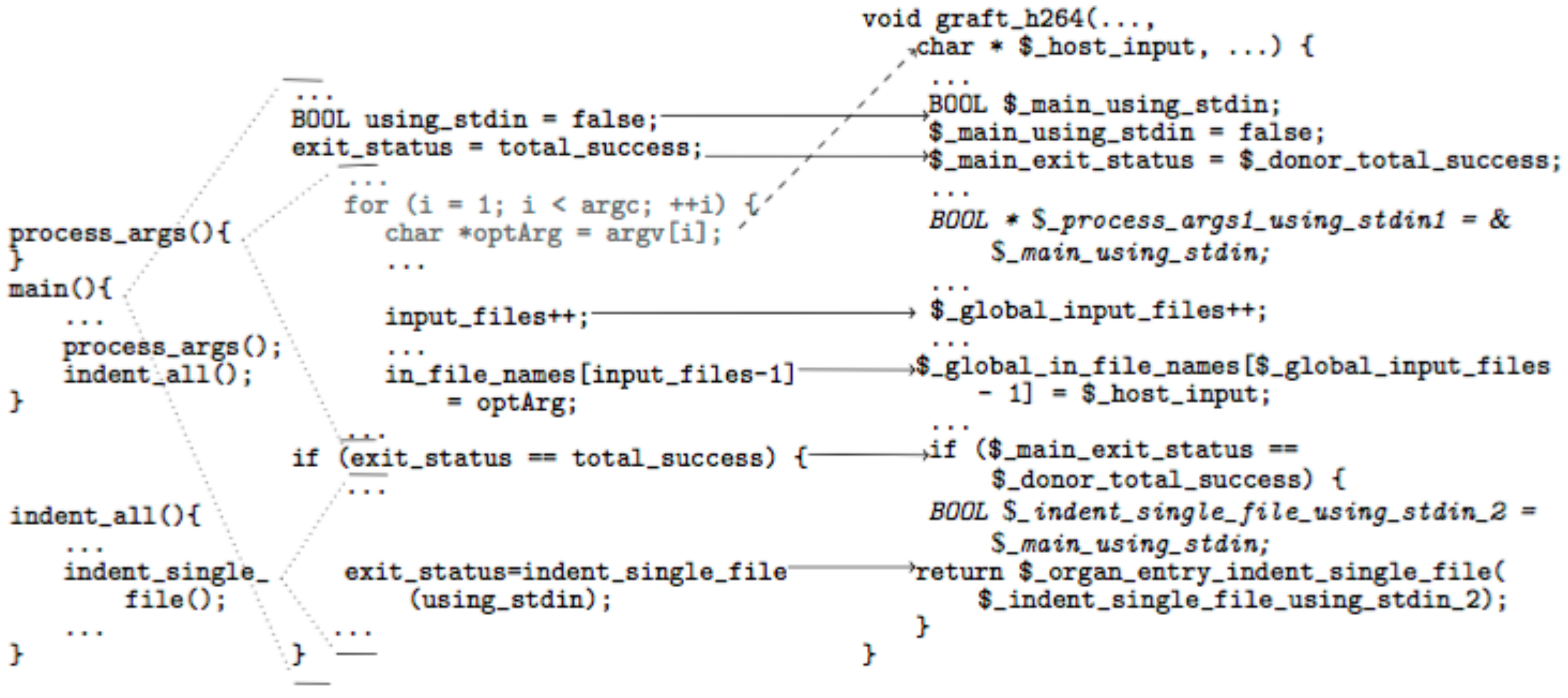
Figure 3. A Module Dependency Graph and its Modularisation using Bunch, taken from [65]

Deep Parameter Optimisation

- Reveal a property hidden in software as a parameter for tuning.
  - F. Wu, W. Weimer, M. Harman, Y. Jia, and J. Krinke. Deep parameter optimisation. In Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO 2015, pages 1375–1382, 2015.
  - J. Sohn, S. Lee, and S. Yoo. Amortised deep parameter optimisation of GPGPU work group size for OpenCV. In F. Sarro and K. Deb, editors, Proceedings of the 8th International Symposium on Search Based Software Engineering, volume 9962 of Lecture Notes in Computer Science, pages 211–217. Springer International Publishing, 2016.



Code Transplantation



```
process_args(){
}
main(){
    ...
    process_args();
    indent_all();
}


indent_all(){
    ...
    indent_single_
        file();
    ...
}
```

(a) Indent Code

```
...
BOOL using_stdin = false;
exit_status = total_success;
  ...
  for (i = 1; i < argc; ++i) {
     char *optArg = argv[i];
     ...

     input_files++;
     ...
     in_file_names[input_files-1]
          = optArg;
  ...
if (exit_status == total_success) {
  ...


  exit_status=indent_single_file
      (using_stdin);
  ...
}
```

(b) Indent Inlined Source Code

```
void graft_h264(...,
    char * $_host_input, ...) {
     ...
     BOOL $_main_using_stdin;
     $_main_using_stdin = false;
     $_main_exit_status = $_donor_total_success;
     ...
     BOOL * $_process_args1_using_stdin1 = &
         $_main_using_stdin;
     ...
     $_global_input_files++;
     ...
    $_global_in_file_names[$_global_input_files
        - 1] = $_host_input;
     ...
     if ($_main_exit_status ==
         $_donor_total_success) {
     BOOL $_indent_single_file_using_stdin_2 =
         $_main_using_stdin;
    return $_organ_entry_indent_single_file(
        $_indent_single_file_using_stdin_2);
    }
}
```

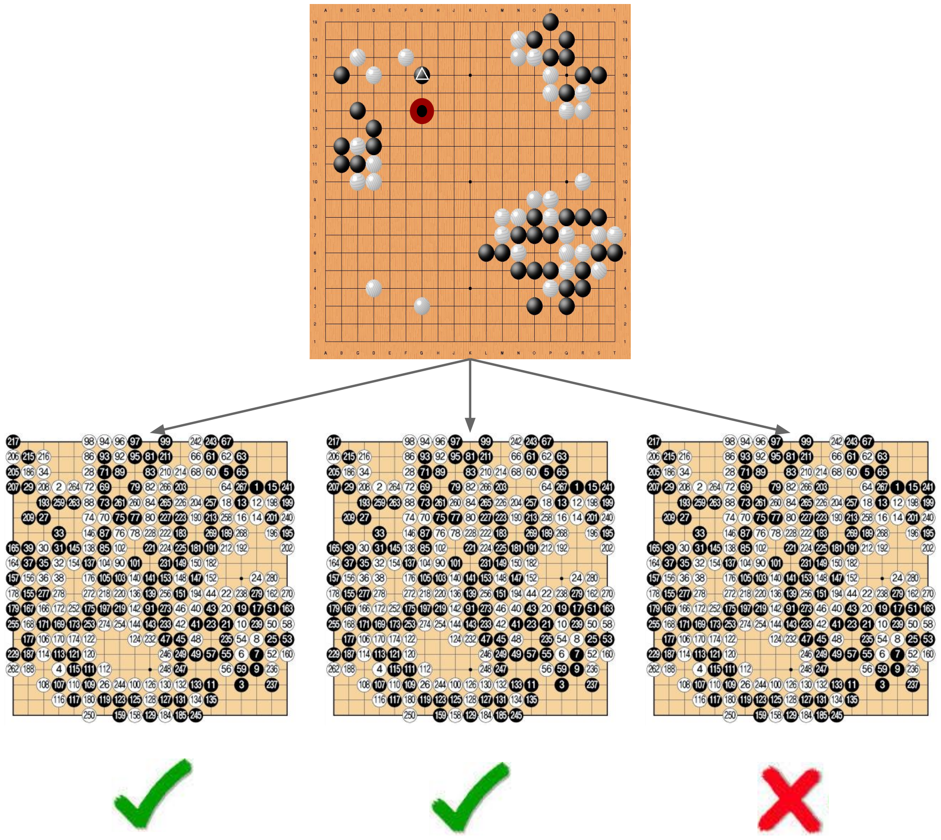
(c) Transplanted Source Code

Fig. 1: Transplant operation in Cflow donor transplant. Code snippet from the beginning of the graft. ···| means function inlining; optArg is mapped to $_host_input; → means original statement replacement under α — renaming; grayed statements are deleted.

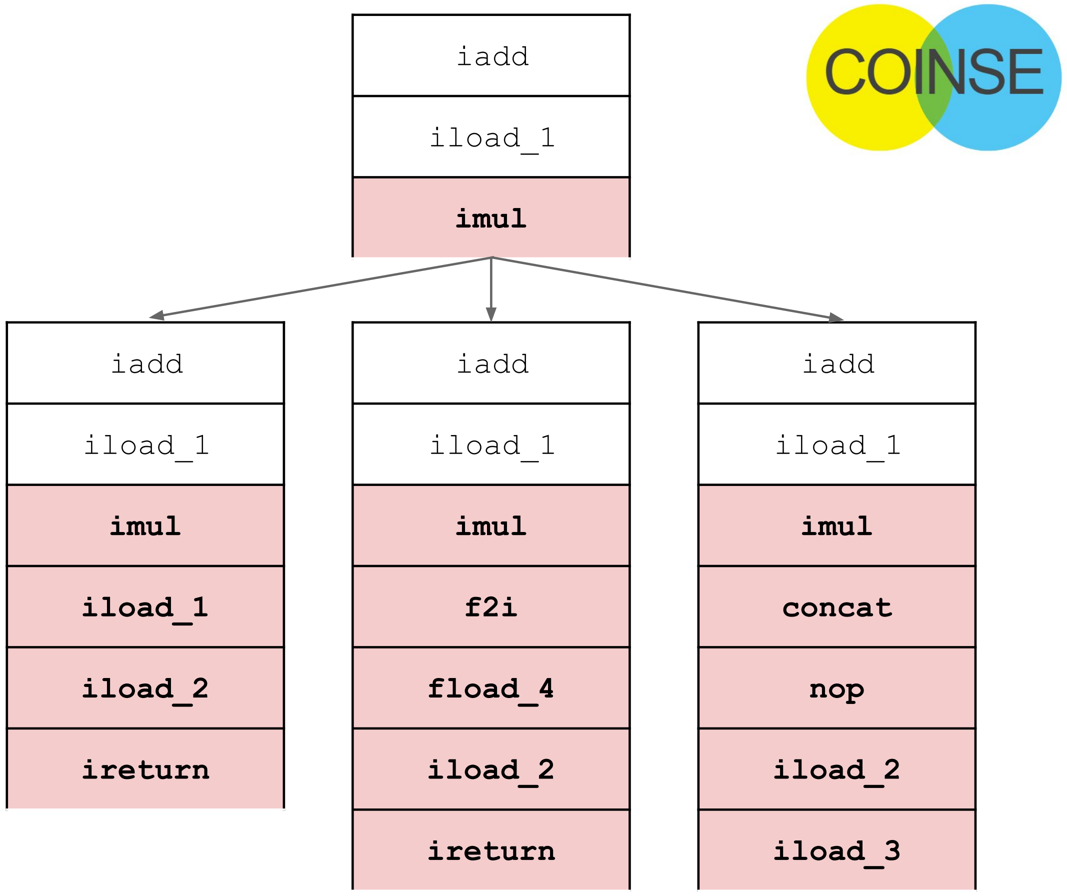
- E. T. Barr, M. Harman, Y. Jia, A. Marginean, and J. Petke. Automated software transplantation. In Proceedings of the 2015 International Symposium on Software Testing and Analysis, ISSTA 2015, pages 257–269, New York, NY, USA, 2015. ACM.
- A. Marginean, E. Barr, M. Harman, and Y. Jia. Automated transplantation of call graph and layout features into kate. In M. Barros and Y. Labiche, editors, Search-Based Software Engineering, volume 9275 of Lecture Notes in Computer Science, pages 262–268. Springer International Publishing, 2015.

Monte Carlo Tree Search for Program Synthesis (instead of GP)

Move *B*



Reward (winning rate) = $\frac{2}{3}$



iadd
iload_1
imul

iadd
iload_1
imul
iload_1
iload_2
ireturn

iadd
iload_1
imul
f2i
fload_4
iload_2
ireturn

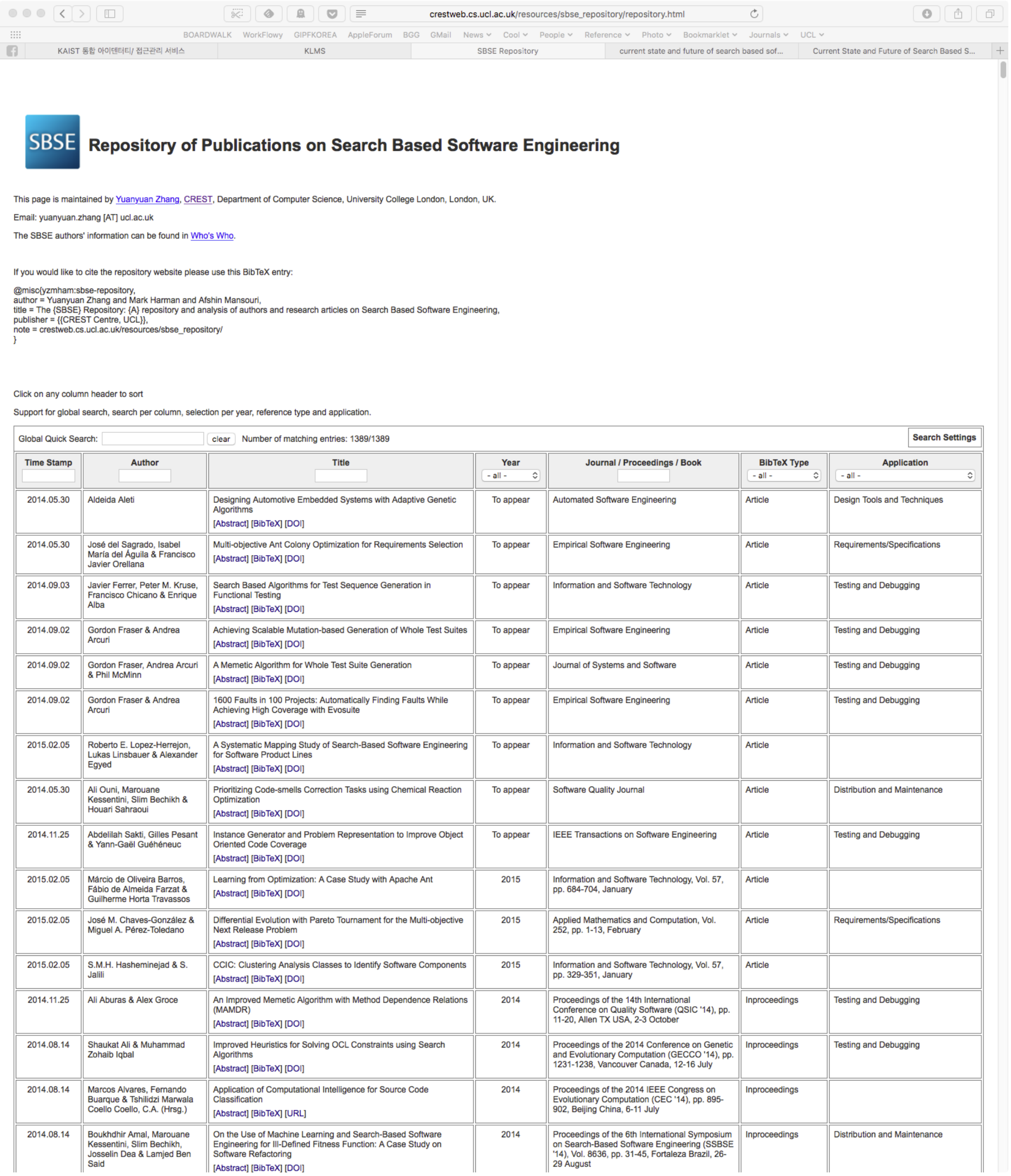
iadd
iload_1
imul
concat
nop
iload_2
iload_3

Fitness: $\frac{2}{3}$ Fitness: $\frac{1}{3}$ Fitness: 0

Reward = $\frac{1}{3}$

SBSE Repository

- Most of the papers published on SBSE, stored and categorised online:
- http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/


crestweb.cs.ucl.ac.uk/resources/sbse_repository/repository.html
BOARDWALK
WorkFlowy
GIPFKOREA
AppleForum
BGG
GMail
News
Cool
People
Reference
Photo
Bookmarklet
Journals
UCL
KAIST 통합 아이덴티티/ 접근관리 서비스
KLMS
SBSE Repository
current state and future of search based sof...
Current State and Future of Search Based S...
SBSE

Repository of Publications on Search Based Software Engineering

This page is maintained by Yuanyuan Zhang, CREST, Department of Computer Science, University College London, London, UK.

Email: yuanyuan.zhang [AT] ucl.ac.uk

The SBSE authors' information can be found in Who's Who.

If you would like to cite the repository website please use this BibTeX entry:

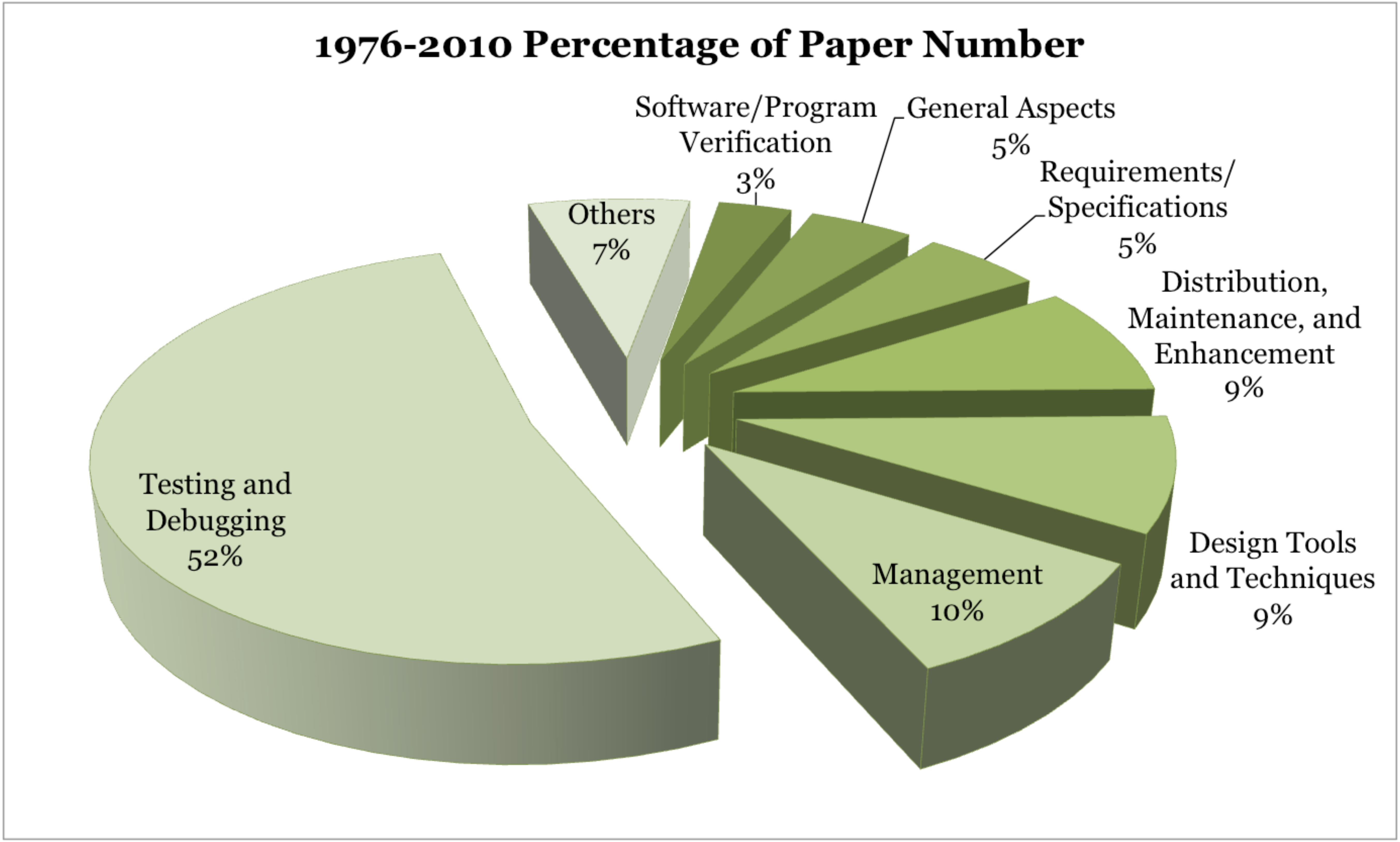
```
@misc{yzmham:sbse-repository,
author = Yuanyuan Zhang and Mark Harman and Afshin Mansouri,
title = The {SBSE} Repository: {A} repository and analysis of authors and research articles on Search Based Software Engineering,
publisher = {{CREST Centre, UCL}},
note = crestweb.cs.ucl.ac.uk/resources/sbse_repository/
}
```

Click on any column header to sort

Support for global search, search per column, selection per year, reference type and application.

Global Quick Search: clear Number of matching entries: 1389/1389 Search Settings

Time Stamp	Author	Title	Year	Journal / Proceedings / Book	BibTeX Type	Application
			- all -		- all -	- all -
2014.05.30	Aldeida Aleti	Designing Automotive Embedded Systems with Adaptive Genetic Algorithms [Abstract] [BibTeX] [DOI]	To appear	Automated Software Engineering	Article	Design Tools and Techniques
2014.05.30	José del Sagrado, Isabel María del Águila & Francisco Javier Orellana	Multi-objective Ant Colony Optimization for Requirements Selection [Abstract] [BibTeX] [DOI]	To appear	Empirical Software Engineering	Article	Requirements/Specifications
2014.09.03	Javier Ferrer, Peter M. Kruse, Francisco Chicano & Enrique Alba	Search Based Algorithms for Test Sequence Generation in Functional Testing [Abstract] [BibTeX] [DOI]	To appear	Information and Software Technology	Article	Testing and Debugging
2014.09.02	Gordon Fraser & Andrea Arcuri	Achieving Scalable Mutation-based Generation of Whole Test Suites [Abstract] [BibTeX] [DOI]	To appear	Empirical Software Engineering	Article	Testing and Debugging
2014.09.02	Gordon Fraser, Andrea Arcuri & Phil McMinn	A Memetic Algorithm for Whole Test Suite Generation [Abstract] [BibTeX] [DOI]	To appear	Journal of Systems and Software	Article	Testing and Debugging
2014.09.02	Gordon Fraser & Andrea Arcuri	1600 Faults in 100 Projects: Automatically Finding Faults While Achieving High Coverage with Evosuite [Abstract] [BibTeX] [DOI]	To appear	Empirical Software Engineering	Article	Testing and Debugging
2015.02.05	Roberto E. Lopez-Herrejon, Lukas Linsbauer & Alexander Egyed	A Systematic Mapping Study of Search-Based Software Engineering for Software Product Lines [Abstract] [BibTeX] [DOI]	To appear	Information and Software Technology	Article	
2014.05.30	Ali Ouni, Marouane Kessentini, Slim Bechikh & Houari Sahraoui	Prioritizing Code-smells Correction Tasks using Chemical Reaction Optimization [Abstract] [BibTeX] [DOI]	To appear	Software Quality Journal	Article	Distribution and Maintenance
2014.11.25	Abdelilah Sakti, Gilles Pesant & Yann-Gaël Guéhéneuc	Instance Generator and Problem Representation to Improve Object Oriented Code Coverage [Abstract] [BibTeX] [DOI]	To appear	IEEE Transactions on Software Engineering	Article	Testing and Debugging
2015.02.05	Márcio de Oliveira Barros, Fábio de Almeida Farzat & Guilherme Horta Travassos	Learning from Optimization: A Case Study with Apache Ant [Abstract] [BibTeX] [DOI]	2015	Information and Software Technology, Vol. 57, pp. 684-704, January	Article	
2015.02.05	José M. Chaves-González & Miguel A. Pérez-Toledano	Differential Evolution with Pareto Tournament for the Multi-objective Next Release Problem [Abstract] [BibTeX] [DOI]	2015	Applied Mathematics and Computation, Vol. 252, pp. 1-13, February	Article	Requirements/Specifications
2015.02.05	S.M.H. Hasheminejad & S. Jalili	CCIC: Clustering Analysis Classes to Identify Software Components [Abstract] [BibTeX] [DOI]	2015	Information and Software Technology, Vol. 57, pp. 329-351, January	Article	
2014.11.25	Ali Aburas & Alex Groce	An Improved Memetic Algorithm with Method Dependence Relations (MAMDR) [Abstract] [BibTeX] [DOI]	2014	Proceedings of the 14th International Conference on Quality Software (QSIC '14), pp. 11-20, Allen TX USA, 2-3 October	Inproceedings	Testing and Debugging
2014.08.14	Shaukat Ali & Muhammad Zohaib Iqbal	Improved Heuristics for Solving OCL Constraints using Search Algorithms [Abstract] [BibTeX] [DOI]	2014	Proceedings of the 2014 Conference on Genetic and Evolutionary Computation (GECCO '14), pp. 1231-1238, Vancouver Canada, 12-16 July	Inproceedings	Testing and Debugging
2014.08.14	Marcos Alvares, Fernando Buarque & Tshilidzi Marwala Coello Coello, C.A. (Hrsg.)	Application of Computational Intelligence for Source Code Classification [Abstract] [BibTeX] [URL]	2014	Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC '14), pp. 895-902, Beijing China, 6-11 July	Inproceedings	
2014.08.14	Boukhdhir Amal, Marouane Kessentini, Slim Bechikh, Josselin Dea & Lamjed Ben Said	On the Use of Machine Learning and Search-Based Software Engineering for Ill-Defined Fitness Function: A Case Study on Software Refactoring [Abstract] [BibTeX] [DOI]	2014	Proceedings of the 6th International Symposium on Search-Based Software Engineering (SSBSE '14), Vol. 8636, pp. 31-45, Fortaleza Brazil, 26-29 August	Inproceedings	Distribution and Maintenance



The ratio of SE research fields that involoved in SBSE.

CLOSE ✕

Hints for Project Ideas

- Your own experience and/or research
- Reading SBSE papers
- Reading SBSE Challenge Track from SSBSE Conference (see conference proceedings from 2013 and onwards)
- Major SE conferences (ICSE, ESEC-FSE, ASE) have sessions dedicated to SBSE

Project Pitch (29/31 October)

- 10 minute sales pitch on what you plan to do.
- **Explicitly** describe the following:
    - Problem
    - How to formulate as search/optimisation
    - How to evaluate, using which data (remember to allow **sufficient** time for data preparation)
- You are welcome to discuss the idea with me beforehand