

Software Measurement (SOEN 6611) Project



Department of Computer Science and Software
Engineering
Gina Cody School of Engineering and Computer
Science
Concordia University

Professor Jinqiu Yang

Team G:

Sareh Farid (27845782)
Seyed Hamed Valiollahi Bayeki (40057506)
Ahmad Memari (40088010)
Sharareh Keshavarzi (40087339)
Hanieh QasemiBoroujeni (40057756)

Contents

- Selected Metrics
- Projects Overview
- Projects' Data for Metrics
- Metrics' Correlations
- Future work

Projects

Project No.	Project Name	Size of the project	Repository	Issue-tracking System	Written in
1	Apache Commons Codec	51K LOC	Git	JIRA	Java
2	Apache Commons Collections	138K LOC	Git	JIRA	Java
3	Apache Commons Configuration	150K LOC	Git	JIRA	Java
4	Apache Maven Doxia	290K LOC	Git	JIRA	Java

Selected Metrics

- **Metric 1: Statement Coverage**

- $(\text{Number of statements executed} / \text{Total number of statements}) * 100$
- Class Wise
- Test coverage metric
- Tool: JaCoCo
- OutputFile: CSV

- **Metric 2: Branch Coverage**

- $(\text{Number of branches executed} / \text{Total number of branches}) * 100$
- Class Wise
- Test coverage metric
- Tool: JaCoCo
- OutputFile: CSV

- **Metric 3: Mutation Score**

- (Number of mutants killed/Total number of mutants) * 100
- Package Wise
- Test-suite effectiveness metric
- Tool: PITest
- OutputFile: CSV

- **Metric 4: McCabe complexity**

- $M = E - N + 2P$
 - E-Number of edges in the graph
 - N-Number of Nodes in the graph
 - P-Number of connected components.
- Class Wise
- Complexity metric
- OutputFile: CSV

- **Metric 5: Code churn**

- Relative code churn = Number of deleted lines + number of added lines
- Version Wise
- Software maintenance metric
- OutputFile: PDF, CSV

- **Metric 6: Post-release defect density**

- Defect density = Number of known defects / Size of the Software(KSLOC)
- Version Wise
- Software quality metric
- Tools: Jira as bug tracker
- OutputFile: CSV

Projects Data for Metrics

Project	Size	Statement Coverage	Branch Coverage	Mutation Score	Average Complexity	Code churn (for latest version)	Post-release Defect Density (for latest version)
Apache Commons Codec	51K LOC	97%	91%	0.84	22	343	0.093469591
Apache Commons Collections	138K LOC	89%	82%	0.49	14	8456	0.132645542
Apache Commons Configuration	150K LOC	87%	83%	0.62	18	18	0.011267733
Apache Maven Doxia	95K LOC	61%	68%	0.43	21	87	0.017140017

Correlation between Metric 1, 2 and 4: Code (Statement and Branch) coverage and McCabe complexity

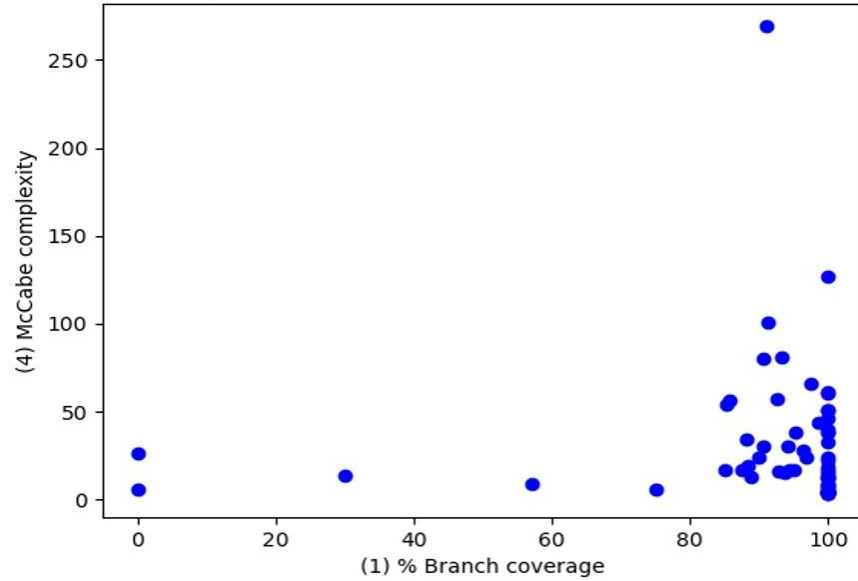
- Branch coverage
 - Branch covered / (branch covered + branch missed)
- Statement coverage:
 - Statement covered / (statement covered + statement missed)
- Complexity coverage
 - Complexity covered + complexity missed

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

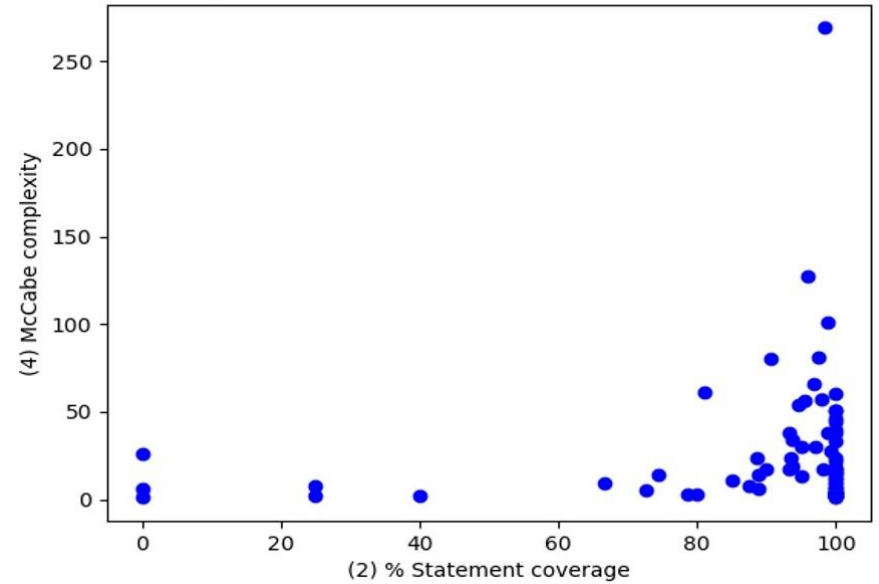
Correlation between 1,2 and 4

Project	Statement coverage (a)	Branch Coverage (b)	Average Cyclomatic complexity (c)	Spearman Correlation (a&c)	Spearman Correlation (b&c)
Apache Commons Codec	97%	91%	22	-0.28	-0.25
Apache Commons Collections	89%	82%	14	-0.33	-0.34
Apache Commons Configuration	87%	83%	18	-0.35	-0.38
Apache Maven Doxia	61%	68%	21	0.03	-0.12

Spearman Correlation R=-0.28 P=0.0271



Spearman Correlation R=-0.25 P=0.0161



Apache Commons Codec project

Future work

- Correlation between 1, 2 and 3: Code (Statement and Branch) coverage and test suite effectiveness
- Correlation between 1, 2 and 6: Code (Statement and Branch) coverage and Post-release defect density
- Correlation between 5 and 6: Code churn and Post-release defect density

The End

Thank you

April 2020

A solid orange horizontal bar spanning the width of the slide at the bottom.