SOEN 6611
SOFTWARE MEASUREMENT
WINTER 2020

Team –N

SWETHA CHENNA
VENKATA PAVAN KUMAR REDDY RAVI
NANDINI BANDLAMUDI
GURMINDER PAL DHIMAN

OUTLINE

1. PROJECTS SELECTED
2. METRICS SELECTED
3. METRICS CALCULATION FOR PROJECTS SELECTED
4. METRIC 1,2 AND 4
5. METRIC 3
6. METRIC 5
7. CORRELATION ANALYSIS
8. FUTURE WORK

PROJECTS SELECTED

1.Apache Commons DBUtils :

- SLOC: 8558

2.Apache Commons Collections :

- SLOC : 118K

3.Apache Commons Configurations :

- SLOC : 122K

METRICS SELECTED

Metric 1 - Statement Coverage:

- This can be considered based on number of statements executed and it is Class level.
- Calculation:(Number of statements executed/Total number of statements)*100

Metric 2 - Branch Coverage :

- This can be considered on number of branches executed and it is Class level
- Calculation:(Number of branches executed/Total number of branches)*100

METRICS SELECTED

Metric 3 - Test Suite Effectiveness :

- This can be considered on mutation score and it is package level
- Calculation : (Number of mutants decreased/Total number of mutants)*100

Metric 4 - Complexity :

- This can be considered on number of linearly independent execution paths and it is Class level
- Calculation : M=E-N+2P
  - E - Number of edges in graph
  - N - Number of nodes in graph
  - P – Number of connected components

METRICS SELECTED

Metric 5 – Maintainability Index :

- This can be considered based on halstead volume , cyclomatic complexity , LOC and commented code .It is project level and measure maintainability of project.
- Calculation :MI=171-(5.2*ln(V)+0.23*(G)+16.2*ln(LOC))
  - V = Halstead Volume
  - G = Cyclomatic Complexity
  - LOC = Count of source lines of code

Metric 6 – Post-release defect density:

- This can be considered based on bugs reported in bug tracker and it is version level .This can be referred as quality indicator of project which considers number of bugs after release
- Calculation :Defect Density = Number of bugs /SLOC

METRICS CALCULATION FOR PROJECTS SELECTED

- 1.Apache Commons DBUtils :
  - SLOC : 8558
  - Code Coverage : 64%
  - Branch Coverage : 77%
  - Mutation coverage : 47%
  - Average Complexity : 7.1
  - Maintainability Index : 79.03
  - Post-release defect density : 0.000874

METRICS CALCULATION FOR PROJECTS SELECTED

- 1.Apache Commons Collections:

- SLOC :118k
- Code Coverage : 51%
- Branch Coverage : 82%
- Mutation coverage: 43%
- McCabe Complexity : 12.3
- Maintainability Index : 61.06
- Post-release defect density : 0.0000789

METRICS CALCULATION FOR PROJECTS SELECTED

- 1.Apache Commons Configurations:

- SLOC : 122K
- Code Coverage : 89%
- Branch Coverage : 83%
- Mutation coverage : 80%
- McCabe Complexity : 14.7
- Maintainability Index : 68.92
- Post-release defect density : 0.000169

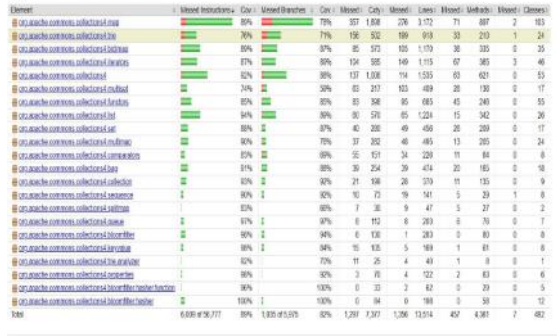
METRIC 1,2 AND 4

Tools /plugin used to calculate the statement and branch coverage i.e., metric 1,2 and 4 is Jacoco(EclEmma) which also gives result of complexity covered.

Apache Commons Configuration

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
org.apache.commons.configuration2.plist		64%		54%	317	635	519	1,479	56	198	1	18
org.apache.commons.configuration2		90%		89%	261	1,646	296	3,619	137	1,008	0	74
org.apache.commons.configuration2.io		78%		74%	72	389	158	879	16	228	1	30
org.apache.commons.configuration2.beanutils		86%		84%	34	233	50	498	3	116	0	9
org.apache.commons.configuration2.resolver		70%		54%	19	53	37	152	1	29	0	4
org.apache.commons.configuration2.interpol		89%		88%	15	117	26	258	9	82	0	13
org.apache.commons.configuration2.convert		95%		90%	28	212	17	420	2	72	0	9
org.apache.commons.configuration2.tree		98%		95%	35	740	14	1,575	5	418	0	48
org.apache.commons.configuration2.builder.combined		97%		93%	19	310	14	771	4	200	0	20
org.apache.commons.configuration2.web		79%		72%	7	34	10	59	3	23	1	6
org.apache.commons.configuration2.tree.xpath		96%		94%	9	151	16	305	3	83	0	9
org.apache.commons.configuration2.reloading		95%		100%	0	85	6	177	0	54	0	9
org.apache.commons.configuration2.event		98%		97%	3	106	4	224	1	68	0	9
org.apache.commons.configuration2.builder		99%		98%	4	303	4	638	1	200	0	24
org.apache.commons.configuration2.ex		85%		n/a	2	12	4	24	2	12	0	3
org.apache.commons.configuration2.builder.fluent		100%		100%	0	61	0	77	0	56	0	3
org.apache.commons.configuration2.sync		100%		100%	0	14	0	23	0	13	0	3
Total	5,705 of 45,968	87%	744 of 4,416	83%	825	5,101	1,175	11,178	243	2,860	3	291

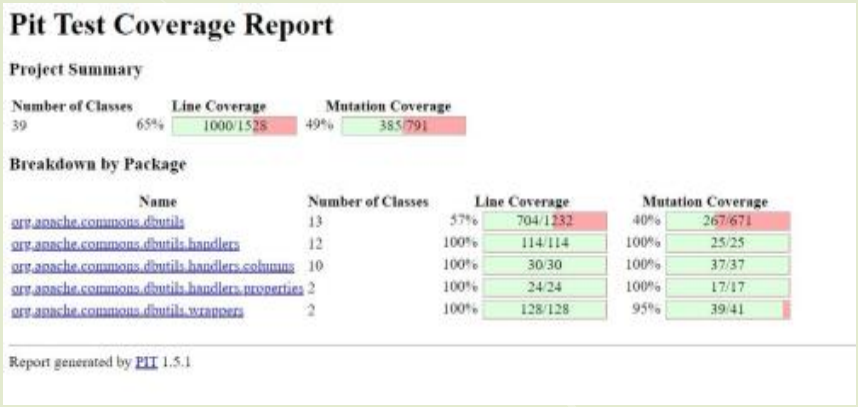
Apache Commons Collections



Apache Commons DbUtils

METRIC 3

Tool /plugin used to calculate the test suite effectiveness i.e., mutation score is maven-pitest plugin



Pit Test Coverage Report

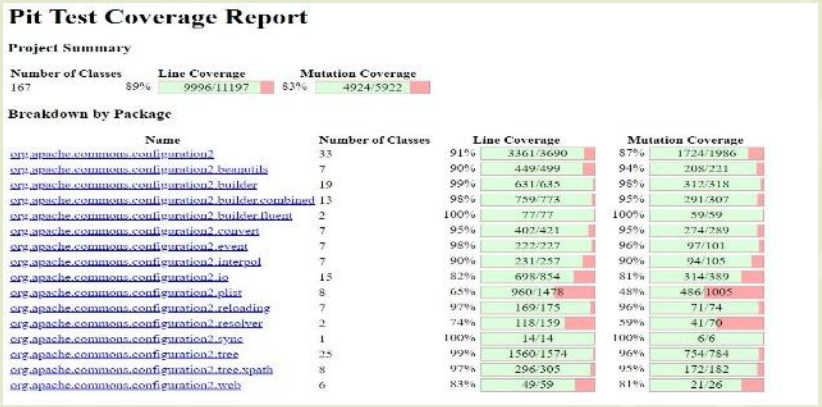
Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
39	65%	1000/1528	49%	385/791

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
org.apache.commons.dbutils	13	57%	704/1232	40%	267/671
org.apache.commons.dbutils.handlers	12	100%	114/114	100%	25/25
org.apache.commons.dbutils.handlers.columns	10	100%	30/30	100%	37/37
org.apache.commons.dbutils.handlers.properties	2	100%	24/24	100%	17/17
org.apache.commons.dbutils.wrappers	2	100%	128/128	95%	39/41

Report generated by PIT 1.5.1



Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage		Mutation Coverage	
167	89%	9996/11197	83%	4924/5922

Breakdown by Package

Name	Number of Classes	Line Coverage		Mutation Coverage	
org.apache.commons.configuration2	33	91%	3361/3690	87%	1724/1986
org.apache.commons.configuration2.beanutils	7	90%	449/499	94%	208/221
org.apache.commons.configuration2.builder	19	99%	631/635	98%	312/318
org.apache.commons.configuration2.builder.combined	13	98%	759/773	95%	291/307
org.apache.commons.configuration2.builder.fluent	2	100%	77/77	100%	59/59
org.apache.commons.configuration2.convert	7	95%	402/421	95%	274/289
org.apache.commons.configuration2.event	7	98%	222/227	96%	97/101
org.apache.commons.configuration2.interpol	7	90%	231/257	90%	94/105
org.apache.commons.configuration2.io	15	82%	698/854	81%	314/389
org.apache.commons.configuration2.jdbc	8	65%	960/1478	48%	486/1005
org.apache.commons.configuration2.reloading	7	97%	169/175	96%	71/74
org.apache.commons.configuration2.resolver	2	74%	118/159	59%	41/70
org.apache.commons.configuration2.sync	1	100%	14/14	100%	6/6
org.apache.commons.configuration2.tree	25	99%	1560/1574	96%	754/784
org.apache.commons.configuration2.tree.xpath	8	97%	296/305	95%	172/182
org.apache.commons.configuration2.web	6	83%	49/59	81%	21/26

METRIC 5

Tool /plugin used to calculate is Jhawk for complexity and maintainability index. JIRA an issue tracker to get bug reports for projects .And we considered API to find the bugs

METRIC 6:

We Used LOCmeter to get the SLOC of the projects and JIRA plugin to obtain the bug report of the particular versions of the projects and calculated the defect density

CORRELATION ANALYSIS

- Correlation analysis between 1,2 and 3
- Correlation analysis between 1,2 and 4
- Correlation analysis between 5 and 6
- Correlation analysis between 1,2 and 6

CORRELATION ANALYSIS BETWEEN 1,2 AND 3

- Observation according to the analysis is that good test suite effectiveness with high code coverage
- Test suites ensures correctness and quality of software.
- Spearman Correlation Coefficient between statement coverage and Mutation Score:
  - Apache Commons DBUtils : 0.3421
  - Apache Commons Collections : 0.82391
  - Apache Commons Configurations : 0.9142

CORRELATION ANALYSIS BETWEEN 1,2 AND 4

- Observation according to the analysis is that when there is high number of branches in source code then complexity will be higher.
- when there is high complexity in classes then its likely to have high coverage test suites
- Spearman Correlation Coefficient between statement coverage and McCabe Complexity:
  - Apache Commons DbUtils : -0.66689
  - Apache Commons Collections : 0.42417
  - Apache Commons Configurations : 0.04871

CORRELATION ANALYSIS BETWEEN 1,2 AND 4

Spearman Correlation Coefficient between Branch coverage and McCabe Complexity:

- Apache Commons DbUtils : -0.5
- Apache Commons Collections : -0.16242
- Apache Commons Configurations : -0.23174

CORRELATION ANALYSIS OF METRIC 5&6

Spearman Correlation Coefficient between maintainability index and Post release defect density:

Projects	Versions	Maintainability Index	Defect density	Spearman coefficient
Apache Commons DbUtils	1.1 1.2 1.4 1.6 1.7	72.63 70.57 78.43 78.79 79.03	0.00173 0.00134 0.000514 0.00074 0.000874	$r_s = -0.5$
Apache Commons Configuration	1.8 2.0 2.2 2.4 2.6	70.41 68.92 72.6 73.92 75.16	0.000379 0.000169 0.000105 0.0000297 0.0000592	$r_s = -0.8$
Apache Commons Collections	2.0 3.0 3.2 4.1 4.4	79.62 78.96 75.24 60.56 61.06	0.000293 0.000326 0.000992 0.000263 0.0000789	$r_s = 0.5$

CORRELATION ANALYSIS OF METRIC 1,2&6

Spearman Correlation Coefficient between statement coverage, Branch coverage and Post release defect density:

Projects	Versions	Statement Coverage	Branch Coverage	Number of bugs	Spearman Coefficient (M1 & M6)	Spearman Coefficient (M2 & M6)
Apache Commons DbUtils	1.1	56%	51%	4	$r_s = -0.22361$	$r_s = 0.22361$
	1.2	63%	53%	4		
	1.4	79%	66%	2		
	1.6	57%	64%	4		
	1.7	64%	77%	6		
Apache Commons Configuration	1.8	77%	70%	16	$r_s = -0.5$	$r_s = -0.5$
	2.0	68%	63%	11		
	2.2	86%	56%	7		
	2.4	79%	77%	2		
	2.6	85%	81%	4		
Apache Commons Collections	2.0	65%	49%	4	$r_s = 0.8$	$r_s = 0.4$
	3.0	76%	68%	15		
	3.2	86%	81%	55		
	4.1	69%	77%	16		
	4.4	51%	82%	5		

THANK YOU