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

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

METRICS CALCULATION FOR PROJECTS SELECTED

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 1	Cov	Missed	City	Missed	Lines	Missed	Methods	Missed:	Classes
grg apache commons configuration2 pist		64%		54%	317	635	519	1,479	56	198	1	18
grg 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	1	70%	1	54%	19	53	37	152	1	29	0	4
org apache commons configuration2 interpol	1	89%	1	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	1	95%	1	100%	0	85	6	177	0	54	0	9
⊕org.apache.commons.configuration2.event	1	98%	1	97%	3	106	4	224	1	68	0	9
grg. apache.commons.configuration2.builder		99%		98%	4	303	4	638	1	200	0	24
∰org.apache.commons.configuration2.ex		85%		nia	2	12	4	24	2	12	0	3
org.apache.commons.configuration2.builder.fluent	1	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

Denest :	Missed Instructions+	Cov	Messed Branches +	Cav	Misso	Coty	Mosed		Histori	Methods:	Missed-	Classes!
■ orpuseche commens sefections4 map		80%		TEN	357	1,808	276	3.172	71	188	- 2	103
copacache commons collectional trie		76%		71%	156	802	199	918	- 33	210	1	- 21
Crts assectio commons polections 4 betimes		20%		375	85	573	105	1,170	38	335		35
COS assectes commons pollections 4 livrators		87%		89%	194	585	149	1,115	67	385	3	46
■ orc. seachs commons, collections/		EN		30%	137	1,00E	754	1,535	63	621	0	. 53
ora assectio commons collections 4 multised	=	34%		59%	- 63	257	103	409	25	130	- 2	17
organische commons polectionst fundurs	=	85%	200	85%	83	338	95	665	45	245	- 0	55
Cots, acache commons, collections 4 led		944		39%	80	571	- 65	1,224	15	342	. 0	.26
COLEXACTO COMPONE COTOCICES 4 set		55%	1	37%	40	200	49	456	26	209	0.	17
grapacite commons collections/, multimac	=	90%	2	TEN	37	382	- 66	485	13	265	- 0	24
E cruesache commons collections à comparation	1	15%	1	67%	爱	151	34	220	- 11	84	- 1	. 8
Croposche commons colections4 bas	=	11%		8%	39	254	39	454	20	165	0	18
■ crouscate commons collections4 collection	=	13%	1	97%	21	198	28	330	55	135	0	9
COLOCACION COM TIONS CONTROL SECUENCE	1	80%	1	10%	10	72	. 9	141	- 5	29	1	8
Crt. ateche commons collections 4 salitmen		EA		6%	7	2	9	47	5	27		2
Cro, mecha commons collections 4 queue	1	67%	1	9%	.6	110	8	203	- 8	76	0	7
COL goache commons colections4 bicomittes		90%	1	98%	- 6	TXE		283	. 0	80	0	8
Crt. mache commons collections Alianymbia	1	16%	1	34%	15	125	- 5	169	- 1	Et.	- 0	8
on anache commons colections4 the analyzer		12%		77%	11	8	4	4)	1.3	8	- 0	t
COLLEGACIO COMMONS CORRECTORES DESCRIPTION		96%		32%	3	n	4	122	- 2	63	- 5	6
Critical and the commons collections a bicomfilter hasher function		36%		100%	. 0	I	2	62		29	- 0	5
Organische commons collections à bicomfilter histher	2	100%	1	100%	0	114	0	198	- 0	58		12
Ices	6.00B of 56.777	19%	1,000 of 5,975	82%	1,297	7,377	1,356	13.554	457	4.361	- 7	482

Apache Commons DbUtils

Bankit I	Missia Mat ultimak	Cov	Missed Branches	Cour	Missed	Cdy	Missed	Lines	Missad	VUSOUS-	Missadi	Classes
# emusione commons durin		28%		72%	793	501	.571	1,151	235	435	T	. 33
# argustatife.commons.dbutis.wapears	E .	160%	=	33%	2	- 66	- 2	123	0	50	- 0	2
# approximate and a maker	1	100%	1	1275	U	58		1114	0	45		12
E crustatie commeschala verlesscharas	1	100%		27%	. 5	- 24		20	0	31	0	- 36
iji orgusaane commons diuris, handlors properties		100%	E	35%		16		- 24	- 0	. 0	- 0	- 1
ldal	2,042 d 5,580	6000	87 6/388	77%	306	764	59.7	1,487	225	570	Y	

Pit Test Coverage Report Project Summary Number of Classes Line Coverage Mutation Coverage 65% 1000/1528 49% 385/791 Breakdown by Package Number of Classes Line Coverage Name Mutation Coverage organache.commons.dbutils 57% 704/1232 267/671 100% 100% 12 114/114 25/25 org.anache.commons.dbutils.handlers 37/37 org.apache.commons.dbutils.handlers.columns 10 100% 30/30 100% ore apache commons dbutils handlers properties 2 100% 24/24 100% 17/17 100% 95% ore anache commons dbutils wrappers 128/128 39/41 Report generated by PIT 1.5.1

METRIC 3

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

Project Summary					
Number of Classes Line Coverage N 167 89% 9996/11197 83%	Iutation Coverage 4924/5922				
Breakdown by Package					
Name	Number of Classes	Lin	ae Coverage	Muta	tion Coverage
orgapache commons configuration2	33	91%	3361/3690	8.7%	1724/1986
org apache commons configuration2 beautifs	7	90%	449/499	94%	208/221
orgapache commons configuration2 builder	19	99%	631/635	98%	312/318
orgapache.commons.configuration2.builder.combine	<u>d</u> 13	98%	759/773	95%	291/307
orgapache commons configuration2 builder fluent	2	100%	77/77	100%	59/59
org apache commons configuration2 convert	7	95%	402/421	95%	274/289
orgapache.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.jo	15	82%	698/854	81%	314/389
org.apache.commons.configuration2.plist	8	65%	960/1478	48%	486/1005
org apache commons configuration? 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

METRIC 6:

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

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

- 显
- 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
	1.1	72.63	0.00173	
Apache Commons	1.2	70.57	0.00134	
Apache Commons DbUtils	1.4	78.43	0.000514	$r_s = -0.5$
DBOtils	1.6	78.79	0.00074	
	1.7	79.03	0.000874	
	1.8	70.41	0.000379	
4	2.0	68.92	0.000169	$r_s = -0.8$
Apache Commons	2.2	72.6	0.000105	r _s = -0.8
Configuration	2.4	73.92	0.0000297	
	2.6	75.16	0.0000592	
	2.0	79.62	0.000293	
0	3.0	78.96	0.000326	
Apache Commons Collections	3.2	75.24	0.000992	$r_{\rm s} = 0.5$
Collections	4.1	60.56	0.000263	
	4.4	61.06	0.0000789	

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)
	1.1	56%	51%	4		
Apache	1.2	63%	53%	4		
Commons	1.4	79%	66%	2	$r_s = -0.22361$	$r_s = 0.22361$
DbUtils	1.6	57%	64%	4		
	1.7	64%	77%	6		
	1.8	77%	70%	16		
Apache	2.0	68%	63%	11		
Commons	2.2	86%	56%	7	$r_{\rm s} = -0.5$	$r_s = -0.5$
Configuration	2.4	79%	77%	2		
	2.6	85%	81%	4		
	2.0	65%	49%	4		
Apache	3.0	76%	68%	15		
Commons	3.2	86%	81%	55	$r_{\rm s} = 0.8$	$r_{\rm s} = 0.4$
Collections	4.1	69%	77%	16		
	4.4	51%	82%	5		

THANK YOU

