Tópicos Avançados em Engenharia de Software

Projeto

Replicando: Evaluating and Improving Semistructured Merge

Aluna: Thayonara de Pontes Alves (tpa@cin.ufpe.br)

Roteiro

- *Projetos escolhidos*
- *Primeiras Observações*
- *Respondendo a RQ1*
- *Respondendo a RQ2*
- *Exemplos das análises*

Projetos escolhidos

Swagger-core - https://github.com/swagger-api/swagger-core

BottomBar - https://github.com/roughike/BottomBar.git

Graphql-java - https://github.com/graphql-java/graphql-java

Truth - https://github.com/google/truth

Volley - https://github.com/google/volley

Servo - https://github.com/Netflix/servo

Bitcoinj - https://github.com/bitcoinj/bitcoinj

Dagger - https://github.com/google/dagger

Xodus - https://github.com/JetBrains/xodus

RxNetty - https://github.com/ReactiveX/RxNetty

Projetos escolhidos

Projeto	Número de commits	Número de colaboradores
Swagger-core	2582	154
Bottom-bar	766	34
GraphQl-java	1166	83
Truth	1074	40
Volley	1975	38
Servo	1237	31
Bitcoinj	2925	84
Dagger	1911	72
Xodus	1466	25
RxNetty	907	26

Projetos escolhidos

	Número de commits	Número de colaboradores
Total	**16011**	**587**

Primeiras observações

Ao analisar um total de **3955 cenários de Merge** de **10 projetos Java**, identificamos **2002 conflitos ao usar a ferramenta de merge não estruturada** e **153 usando a semi-estruturada.** Isso representa uma redução da semi-estruturada de aproximadamente **93% no número total de conflitos relatados**.

Primeiras observações

Como cada cenário de merge pode ter conflitos reportados por ambas as ferramentas, observamos **que os 2002 conflitos reportados por ferramenta de merge não estruturada** ocorreram em **3% dos cenários de merges da amostras**. Além disso, **os 153 conflitos relatados pela ferramenta de merge semi-estruturada** ocorreram em **1% dos cenários de merges da amostras.**

RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***

Ao usar uma ferramenta de merge não estruturada, **4,14% ± 3,77% dos cenários de merges possuem pelo menos um falso positivo adicional UN(aFP (UN)).** Além disso, **29,6% ± 13,5% dos conflitos relatados são falsos positivos aFP(UN).**

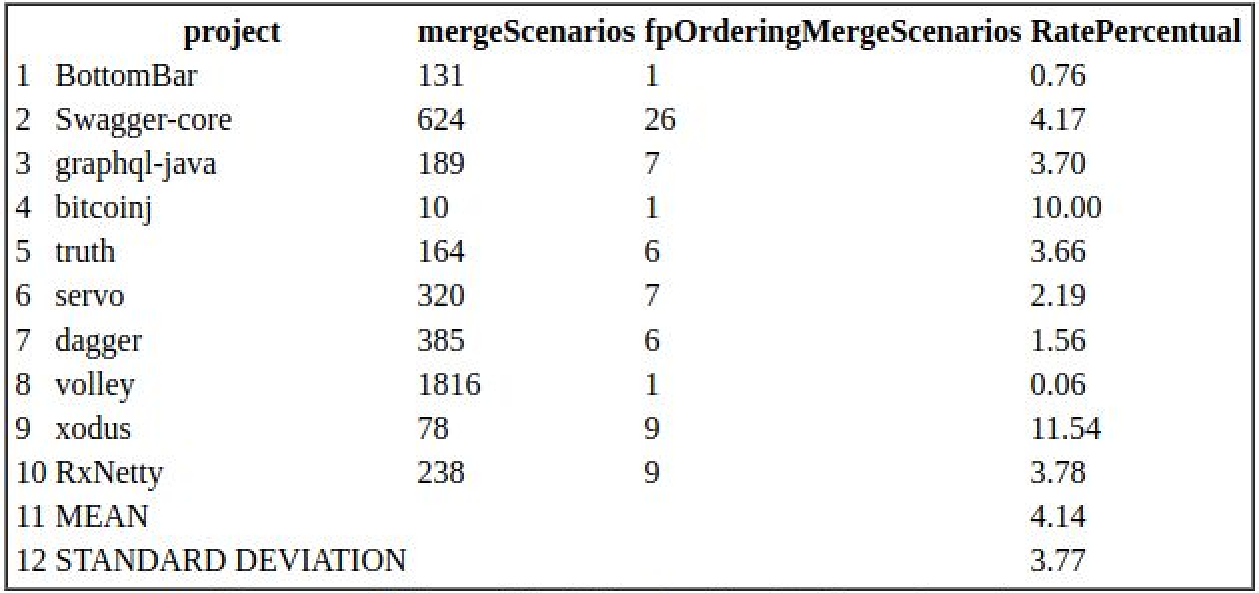
RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***

Isso é maior do que a porcentagem de **(aFP (SS)): 0,98% ± 0,80%.** Além disso, apenas **34% ± 25% dos cenários de merges possuem pelo menos um (aFP (SS)).**

RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***

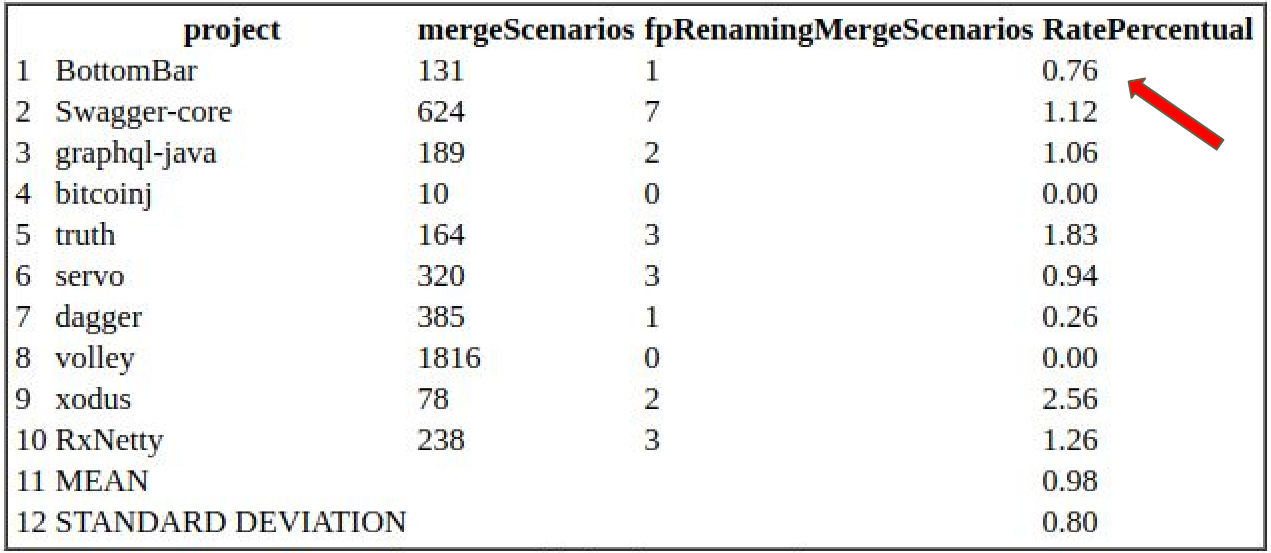


	project	mergeScenarios	fpOrderingMergeScenarios	RatePercentual
1	BottomBar	131	1	0.76
2	Swagger-core	624	26	4.17
3	graphql-java	189	7	3.70
4	bitcoinj	10	1	10.00
5	truth	164	6	3.66
6	servo	320	7	2.19
7	dagger	385	6	1.56
8	volley	1816	1	0.06
9	xodus	78	9	11.54
10	RxNetty	238	9	3.78
11	MEAN			4.14
12	STANDARD DEVIATION			3.77

Unstructured Merge Added False Positives by Merge Scenarios

RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***



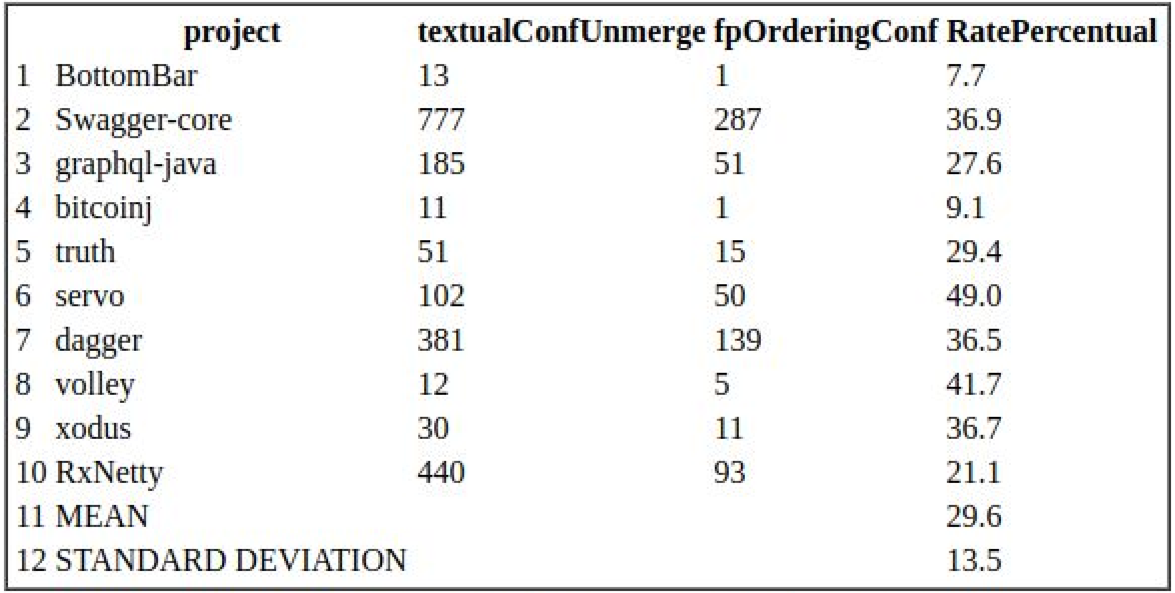
	project	mergeScenarios	fpRenamingMergeScenarios	RatePercentual
1	BottomBar	131	1	0.76
2	Swagger-core	624	7	1.12
3	graphql-java	189	2	1.06
4	bitcoinj	10	0	0.00
5	truth	164	3	1.83
6	servo	320	3	0.94
7	dagger	385	1	0.26
8	volley	1816	0	0.00
9	xodus	78	2	2.56
10	RxNetty	238	3	1.26
11	MEAN			0.98
12	STANDARD DEVIATION			0.80

Semistructured Merge Added False Positives by Merge Scenarios

- *Bottom-Bar tem o mesmo número de aFP(UN) e aFP(SS)*

RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***

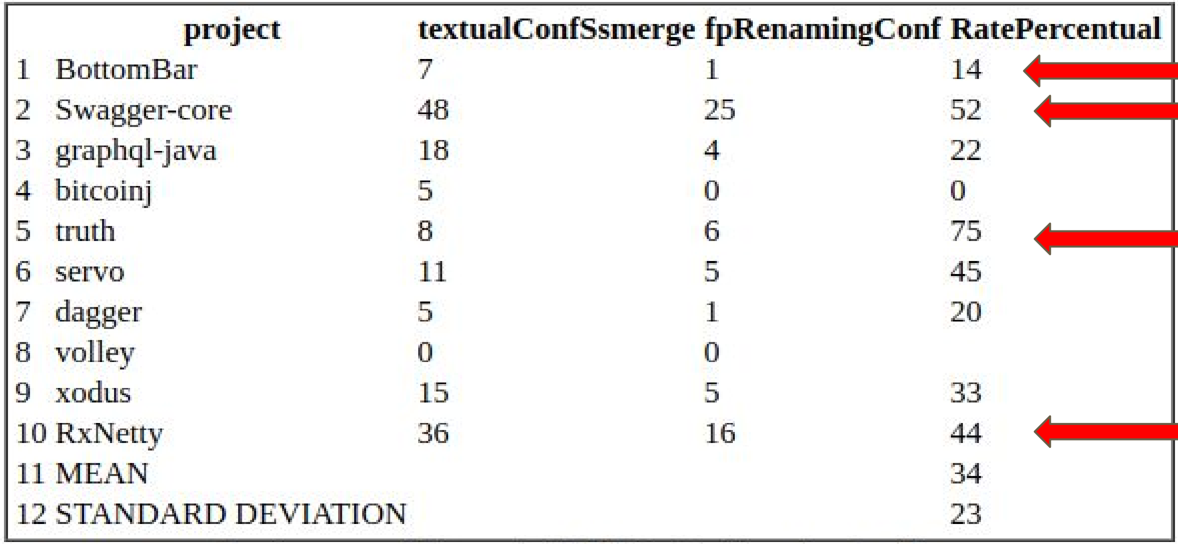


	project	textualConfUnmerge	fpOrderingConf	RatePercentual
1	BottomBar	13	1	7.7
2	Swagger-core	777	287	36.9
3	graphql-java	185	51	27.6
4	bitcoinj	11	1	9.1
5	truth	51	15	29.4
6	servo	102	50	49.0
7	dagger	381	139	36.5
8	volley	12	5	41.7
9	xodus	30	11	36.7
10	RxNetty	440	93	21.1
11	MEAN			29.6
12	STANDARD DEVIATION			13.5

Unstructured Merge Added False Positives by Conflicts

RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***



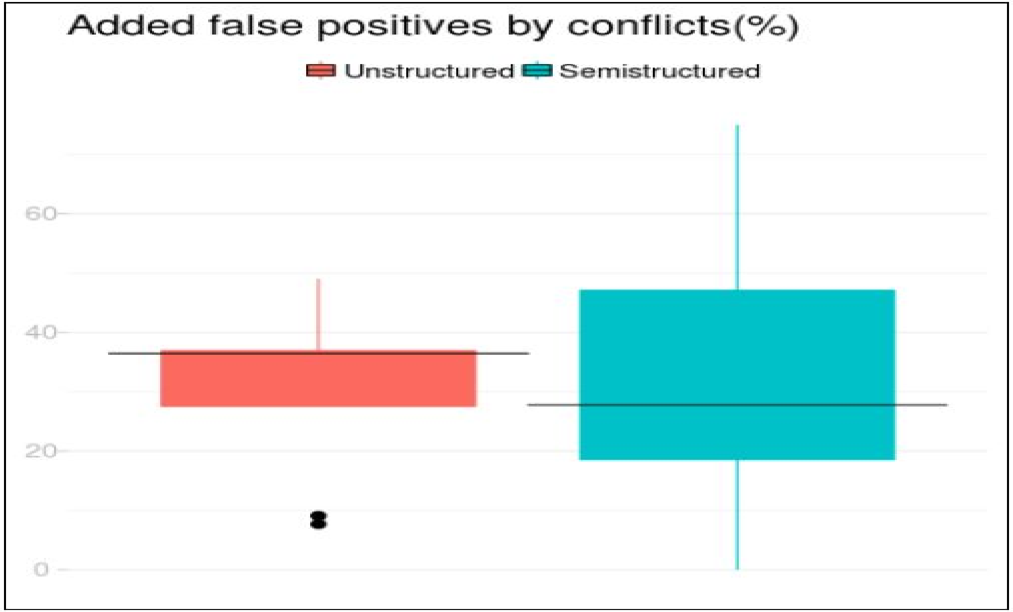
	project	textualConfSsmerge	fpRenamingConf	RatePercentual
1	BottomBar	7	1	14
2	Swagger-core	48	25	52
3	graphql-java	18	4	22
4	bitcoinj	5	0	0
5	truth	8	6	75
6	servo	11	5	45
7	dagger	5	1	20
8	volley	0	0	
9	xodus	15	5	33
10	RxNetty	36	16	44
11	MEAN			34
12	STANDARD DEVIATION			23

Semistructured Merge Added False Positives by Conflicts

BottomBar, Swagger-core, Truth e RxNetty têm mais aFP(SS) do que aFP(UN)

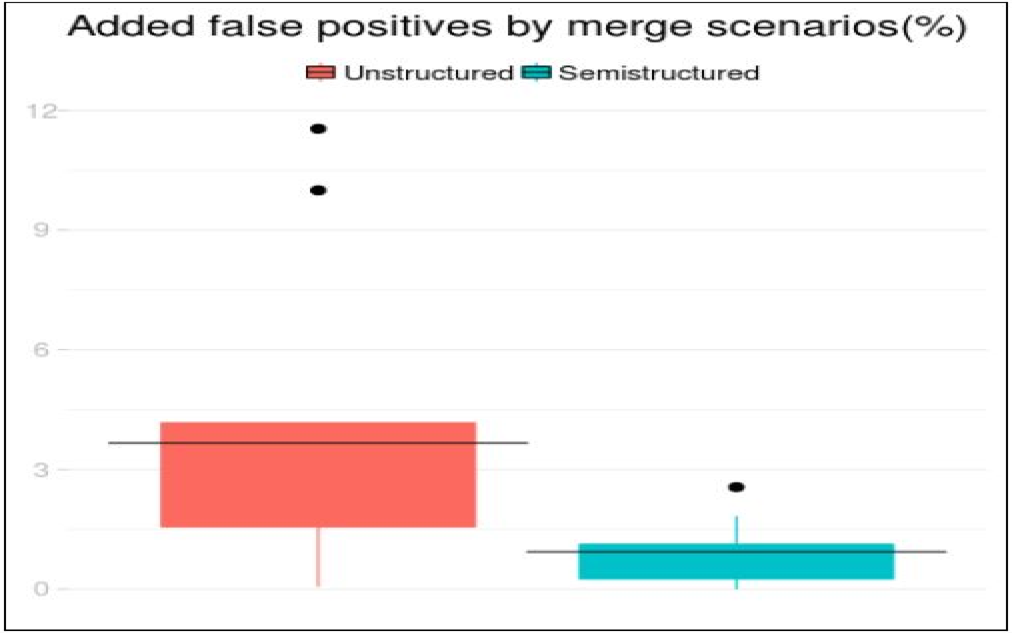
RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***



RQ1

- ***When compared to unstructured merge, does semistructured merge reduce unnecessary integration effort by reporting fewer false positives?***



RQ1

- *Additional False Positives of Semistructured Merge are Easier to Analyze and Resolve.*

```
/home/paper219/Desktop/analysis/demonstration/projects/graphql-java/revisions/rev_9d94784_8f1681d/rev_9d94784-8f1681d/src/main/java/graphql/execution/ExecutionContext.java;<<<<<<< MINE
    private final ExecutionId executionId;
    private final ExecutionStrategy executionStrategy;
||||||| BASE
    private final ExecutionStrategy executionStrategy;
=======
    private final ExecutionStrategy queryStrategy;
    private final ExecutionStrategy mutationStrategy;
>>>>>>> YOURS
```

RQ1

- ***Additional False Positives of Semistructured Merge are Easier to Analyze and Resolve.***

Crosscutting Conflict

RQ1

- ***Additional False Positives of Semistructured Merge are Easier to Analyze and Resolve.***

Ao tentar analisar o código ***de aFP(UN) descobrimos que 44% dos conflitos são crosscutting conflict;*** ou seja, não conseguimos analisar o texto de conflito porque não corresponde a um único elemento de linguagem válido.

RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

Ao usar uma ferramenta de merge SS, **5,13% ± 6,18% dos cenários de merges possuem pelo menos um falso negativo adicional SS(aFN (SS)).** Além disso, **78% ± 27% dos conflitos relatados são falsos negativo aFN(SS).**

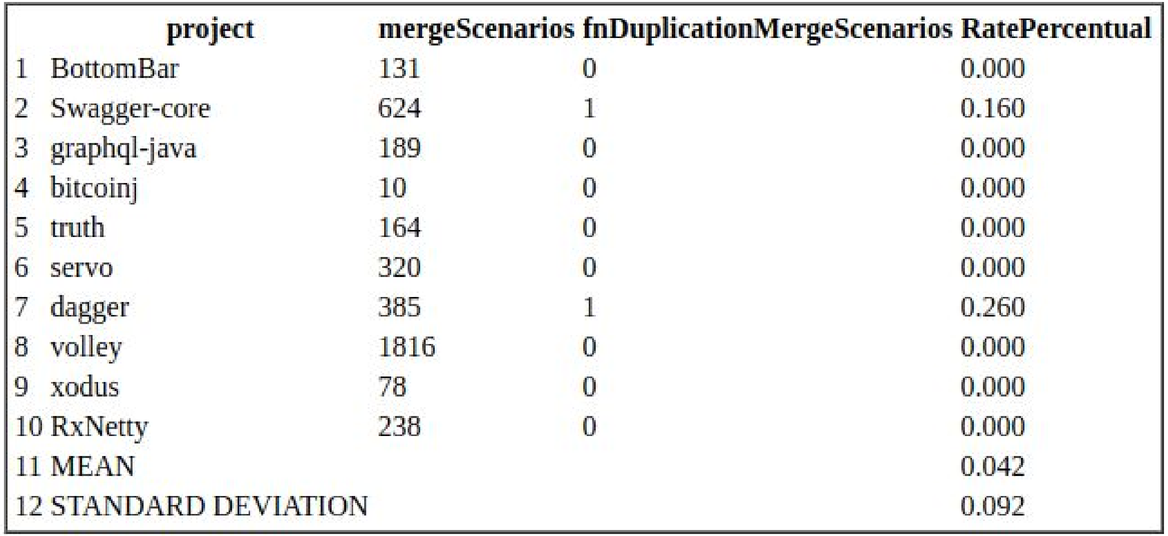
RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

Isso é maior do que a porcentagem de **(aFN (UN)): 0,042% ± 0,092%.** Além disso, apenas **0,065% ± 0,165% dos cenários de merges possuem pelo menos um (aFN (UN)).**

RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

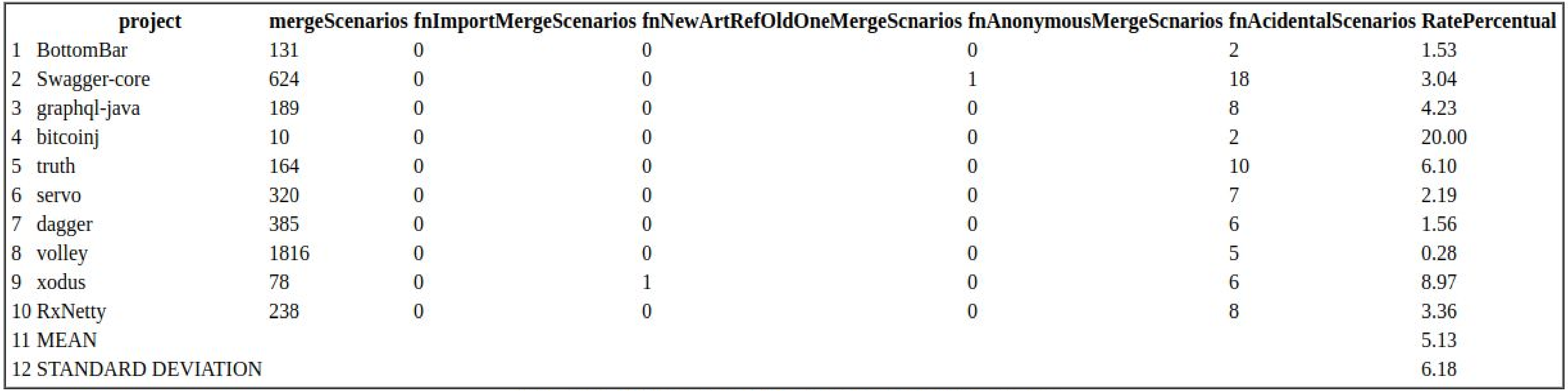


	project	mergeScenarios	fnDuplicationMergeScenarios	RatePercentual
1	BottomBar	131	0	0.000
2	Swagger-core	624	1	0.160
3	graphql-java	189	0	0.000
4	bitcoinj	10	0	0.000
5	truth	164	0	0.000
6	servo	320	0	0.000
7	dagger	385	1	0.260
8	volley	1816	0	0.000
9	xodus	78	0	0.000
10	RxNetty	238	0	0.000
11	MEAN			0.042
12	STANDARD DEVIATION			0.092

Unstructured Merge Added False Negatives by Merge Scenarios

RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

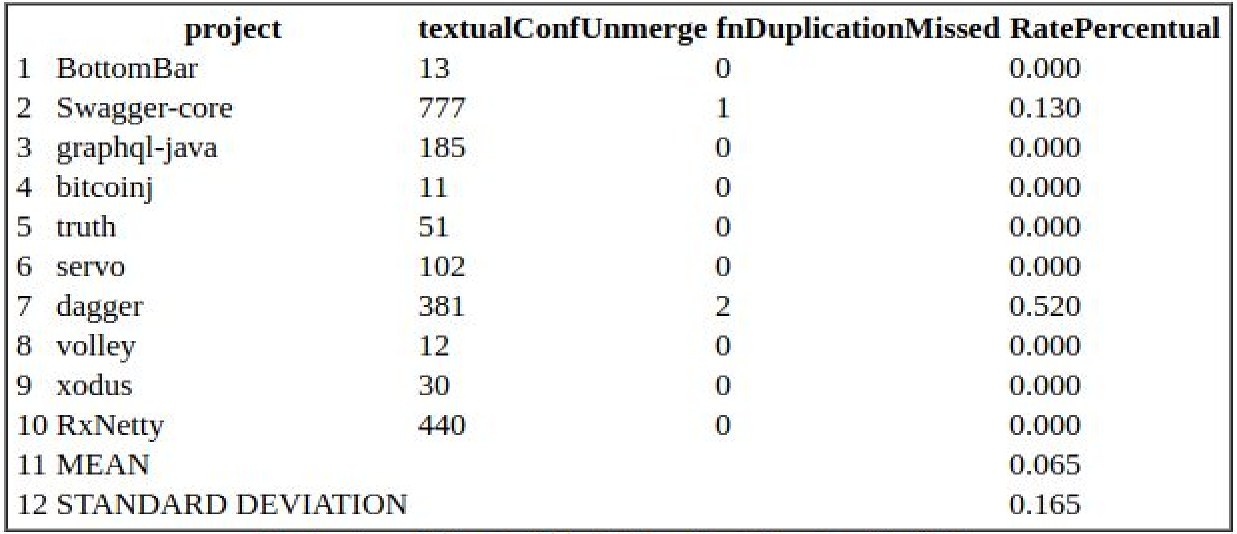


	project	mergeScenarios	fnImportMergeScenarios	fnNewArtRefOldOneMergeScnarios	fnAnonymousMergeScnarios	fnAcidentalScenarios	RatePercentual
1	BottomBar	131	0	0	0	2	1.53
2	Swagger-core	624	0	0	1	18	3.04
3	graphql-java	189	0	0	0	8	4.23
4	bitcoinj	10	0	0	0	2	20.00
5	truth	164	0	0	0	10	6.10
6	servo	320	0	0	0	7	2.19
7	dagger	385	0	0	0	6	1.56
8	volley	1816	0	0	0	5	0.28
9	xodus	78	0	1	0	6	8.97
10	RxNetty	238	0	0	0	8	3.36
11	MEAN						5.13
12	STANDARD DEVIATION						6.18

Semistructured Merge Added False Negatives by Merge Scenarios

RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

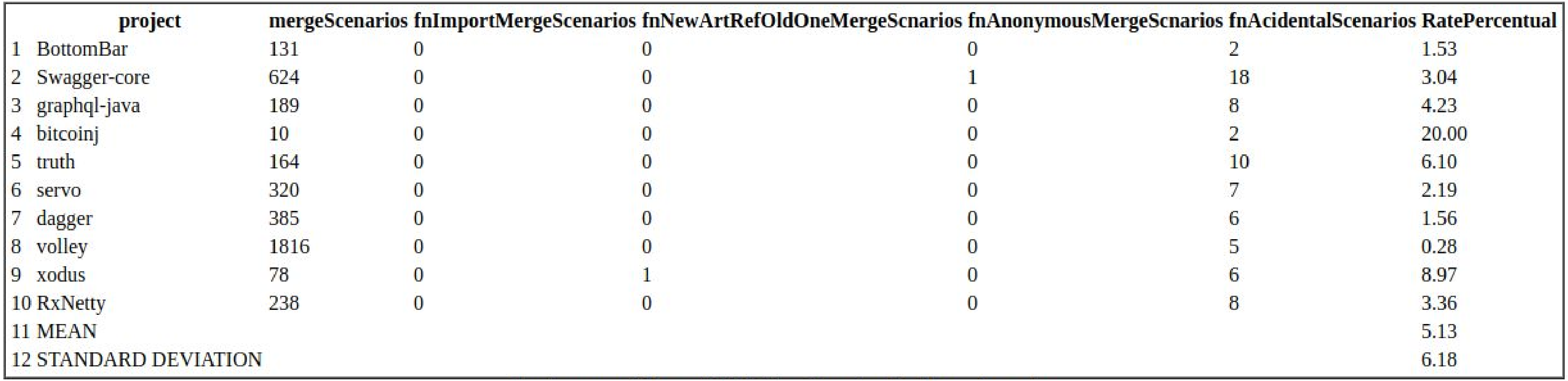


	project	textualConfUnmerge	fnDuplicationMissed	RatePercentual
1	BottomBar	13	0	0.000
2	Swagger-core	777	1	0.130
3	graphql-java	185	0	0.000
4	bitcoinj	11	0	0.000
5	truth	51	0	0.000
6	servo	102	0	0.000
7	dagger	381	2	0.520
8	volley	12	0	0.000
9	xodus	30	0	0.000
10	RxNetty	440	0	0.000
11	MEAN			0.065
12	STANDARD DEVIATION			0.165

Unstructured Merge Added False Negatives by Conflicts

RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

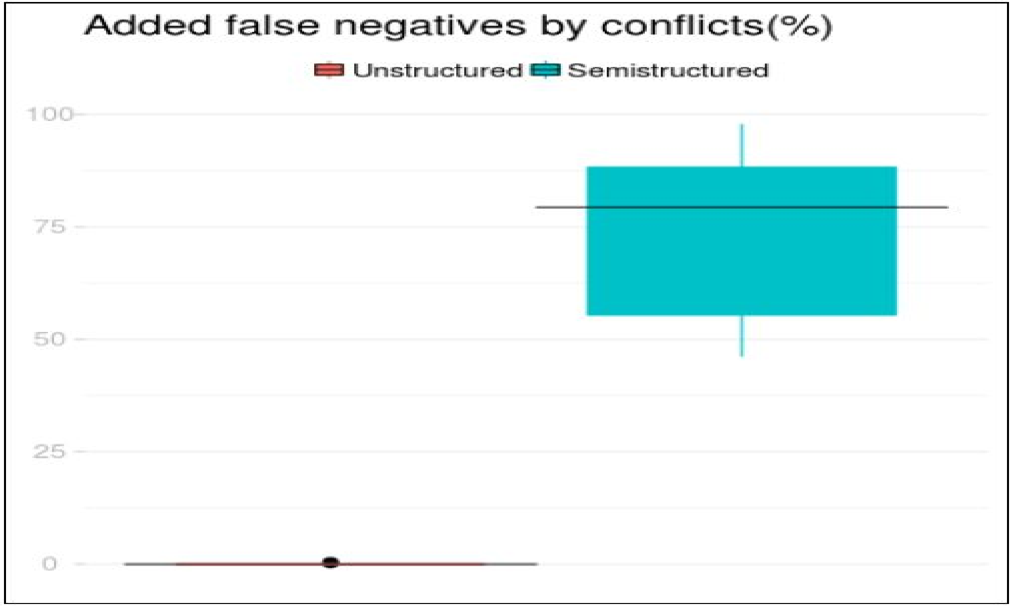


	project	mergeScenarios	fnImportMergeScenarios	fnNewArtRefOldOneMergeScnarios	fnAnonymousMergeScnarios	fnAcidentalScenarios	RatePercentual
1	BottomBar	131	0	0	0	2	1.53
2	Swagger-core	624	0	0	1	18	3.04
3	graphql-java	189	0	0	0	8	4.23
4	bitcoinj	10	0	0	0	2	20.00
5	truth	164	0	0	0	10	6.10
6	servo	320	0	0	0	7	2.19
7	dagger	385	0	0	0	6	1.56
8	volley	1816	0	0	0	5	0.28
9	xodus	78	0	1	0	6	8.97
10	RxNetty	238	0	0	0	8	3.36
11	MEAN						5.13
12	STANDARD DEVIATION						6.18

Semistructured Merge Added False Negatives by Merge Scenarios

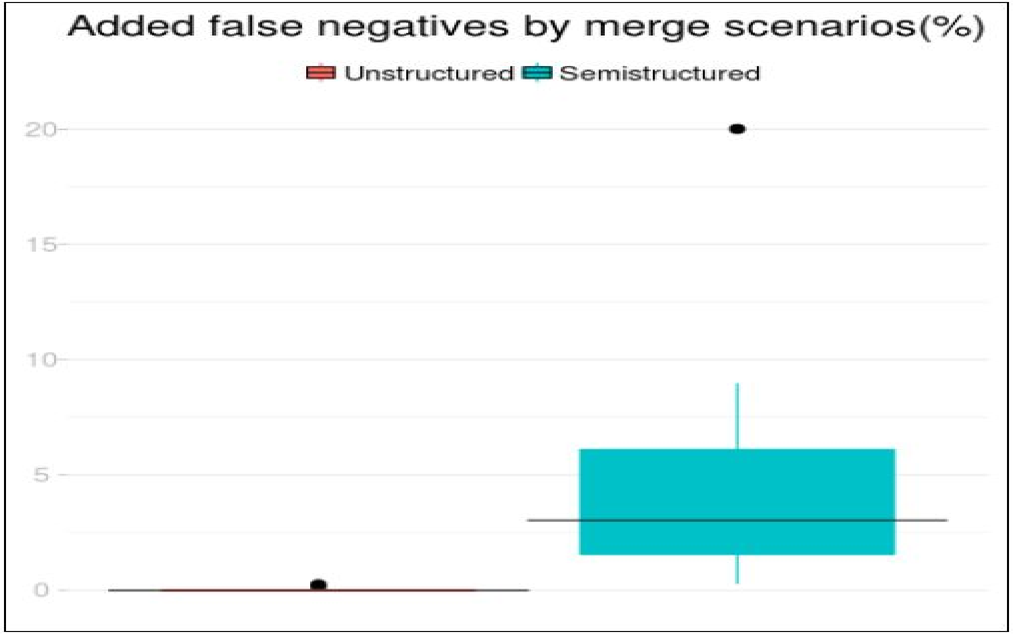
RQ2

- *When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?*



RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***



RQ2

- ***When compared to unstructured merge, does semistructured merge compromise integration correctness by having more false negatives?***

Dos **27 casos analisados**, apenas **3 casos foram confirmados como sendo um aFN(SS).**

RQ2

- *4.2.1 Additional False Negatives of Semistructured Merge are Harder To Detect and Resolve*

```
/home/paper219/Desktop/analysis/demonstration/projects/dagger/revisions/rev_1bc7c83_8f2e49e/rev_1bc7c83-8f2e49e/compiler/src/main/java/dagger/internal/codegen/InjectAdapterProcessor.java;<<<<<<< MINE
    writeSupertypeInjectorField(writer, type, supertype);
||||||| BASE
    writer.emitField(JavaWriter.type(Binding.class,
        rawTypeToString(supertype, '.')), "supertype", PRIVATE);
=======
    writeSupertypeInjectorField(writer, supertype);
>>>>>>> YOURS
```

Exemplos das Análises

/home/paper219/Desktop/analysis/demonstration/projects/truth/revisions/rev_d7a2ff6_db7ed46/rev_d7a2ff6-db7ed46/src/main/java/org/junit/contrib/truth/subjects/ListSubject.java;

```
<<<<<<<  MINE
@GwtCompatible
|||||||  BASE
=======
import  org.junit.contrib.truth.FailureStrategy;
>>>>>>>  YOURS
```

Exemplos das Análises

Mudança : Left adiciona annotation e Right adiciona um import.
Categoria : aFP(UN)
Justificativa : As mudanças não são conflitantes, pode pôr as duas sem problemas. Não causarão comportamentos indesejados a um dos desenvolvedores.
Como resolver: O local de se pôr o import é mais acima na classe, annotation provavelmente em cima de um método. O integrador só deveria se atentar a isso pra ver a melhor forma de integrar.