

Table 1– Measurements results for the affected aspects before and after applying AOR methods: Telecom project

Aspect- Oriented Refactoring Method		Aspect Name	Inheritance		Coupling		Cohesion	Size and Complexity		
			DIT	NOC	CBC	RFC	LCOO	WOC	NOA	LOC
Extract Method form Advice	Before	TimerLog.java	1	0	1	2	0	2	2	12
	After	TimerLog.java	1	0	1	3	1	3	2	15
Introduce Aspect Protection	Before	Billing.java	1	0	4	8	5	7	3	26
	After	Billing.java	1	0	4	8	5	7	4	29
Delete Unreachable Advice	Before	Timing.java	1	0	2	5	3	5	5	23
	After	Timing.java	1	0	2	4	1	4	5	20
Replace Statement List in Advice with Method call	Before	TimerLog.java	1	0	1	3	1	3	3	15
	After	TimerLog.java	1	0	2	4	1	3	3	13
Eliminating Borrowed Pointcut		Billing.java	1	0	4	8	5	7	3	29
	Before	Timing.java	1	0	2	5	3	5	1	23
		TimerLog.java	1	0	2	4	1	3	3	13
		NewTiming.java	-	-	-	-	-	-	-	-
		Billing.java	1	0	3	8	5	7	2	27
	After	Timing.java	1	0	2	4	3	5	1	23
		TimerLog.java	1	0	2	4	1	3	3	13
		NewTiming.java	1	0	3	0	0	0	1	4

Table 1– Continued

Aspect- Oriented Refactoring Method	Aspect Name	Inheritance		Coupling		Cohesion	Size and Complexity			
		DIT	NOC	CBC	RFC	LCOO	WOC	NOA	LOC	
Pull Up Inter-Type Declaration	Before	Billing.java	2	0	3	7	3	6	3	25
		Timing.java	2	0	2	3	2	4	2	21
		CustomerAspect.java	1	2	1	1	0	1	1	5
	After	Billing.java	2	0	2	6	0	5	3	22
		Timing.java	2	0	2	2	0	3	2	19
		CustomerAspect.java	1	2	1	2	1	2	1	8

Table 2– Changes in the internal quality metrics caused by applying refactoring methods: Telecom project

Aspect-Oriented Construct	Aspect- Oriented Refactoring Method	Inheritance		Coupling		Cohesion	Size and Complexity		
		DIT	NOC	CBC	RFC	LCOO	WOC	NOA	LOC
Advice	Extract Method from Advice	-	-	-	↑	↑	↑	-	↑
	Delete Unreachable Advice	-	-	-	↓	↓	↓	-	↓
	Replace Statement List in advice with Method call	-	-	-	-	-	-	-	↓
Declare Clause	Introduce Aspect Protection	-	-	-	-	-	-	↑	↑
Inter-Type Declaration	Pull Up Inter-Type Declaration	-	-	↓	↓	↓	↓	-	↓
Pointcut	Eliminate Borrowed Pointcut	-	-	↓	-	-	-	↓	↓