

# RQ1: Can we quantify interest of TD at the functional level? How much is the interest? (Version 3)

*Yasutaka Kamei*

*Feb 13th, 2016*

## What did we revise (Version 3)?

- A current version shows the summary using only positive interest or negative interest, while the previous version showed the summary using mixed positive and negative interest.

## What did we revise (Version 2)?

- We divide one whole dataset into three projects (jruby, Ant and jmeter)
- To calculate interest, we use the ratio of metrics value of v2 in it of v1. The positive value means we may need to spend additional cost.
  - For division, we exclude the technical debt that has 0 in v1. We show the number of the excluded technical debt.
  - Previous report measured the interest by subtracting the metric value of v2 - the metric value v1.
- We use only one of duplicate technical debt that has same function name and same introducing version

## Data Overview

```
setwd("/Users/kamei/Research/techdebt/msr16_td_interest/")
source("./r_scripts/data_read.r")

fc <- factor(data$Project)

data = cbind(year = (as.Date(data$v2_date) - as.Date(data$v1_date)) / 365, data)

# the number of technical debt at the method-level
tapply(data$Project, fc, length)
```

```
##      apache-ant apache-jmeter      jruby
##           97           234           506
```

```
# choose one of duplicated method and version name
method_and_version_name <- paste(data$Method_Signature, data$v1, sep="")
sum(duplicated(method_and_version_name))
```

```
## [1] 140
```

```
data <- data[!duplicated(method_and_version_name), ]
fc <- factor(data$Project)
tapply(data$Project, fc, length)
```

```
##      apache-ant apache-jmeter      jruby
##           84           180           433
```

```
# any correlation?
```

```
cor(data[,c("CountInput_v1", "CountOutput_v1", "CountLine_v1", "Cyclomatic_v1", "MaxNesting_v1")], method="spearmanr")
```

```
##           CountInput_v1 CountOutput_v1 CountLine_v1 Cyclomatic_v1
## CountInput_v1      1.0000000      0.7709514      0.7739577      0.7863770
## CountOutput_v1     0.7709514      1.0000000      0.8908325      0.8452407
## CountLine_v1       0.7739577      0.8908325      1.0000000      0.8907939
## Cyclomatic_v1      0.7863770      0.8452407      0.8907939      1.0000000
## MaxNesting_v1      0.7361987      0.7938156      0.8305780      0.9371390
##           MaxNesting_v1
## CountInput_v1      0.7361987
## CountOutput_v1     0.7938156
## CountLine_v1       0.8305780
## Cyclomatic_v1      0.9371390
## MaxNesting_v1      1.0000000
```

```
cor(data[,c("CountInput_v2", "CountOutput_v2", "CountLine_v2", "Cyclomatic_v2", "MaxNesting_v2")], method="spearmanr")
```

```
##           CountInput_v2 CountOutput_v2 CountLine_v2 Cyclomatic_v2
## CountInput_v2      1.0000000      0.6985802      0.7197771      0.7191144
## CountOutput_v2     0.6985802      1.0000000      0.8717805      0.8219677
## CountLine_v2       0.7197771      0.8717805      1.0000000      0.8857751
## Cyclomatic_v2      0.7191144      0.8219677      0.8857751      1.0000000
## MaxNesting_v2      0.6749945      0.7753973      0.8135089      0.9286364
##           MaxNesting_v2
## CountInput_v2      0.6749945
## CountOutput_v2     0.7753973
## CountLine_v2       0.8135089
## Cyclomatic_v2      0.9286364
## MaxNesting_v2      1.0000000
```

## Observation

- 140 technical debt is removed due to duplication
- apache-ant has 84 technical debt. The number may be small.
- The following pairs have more than 0.8 correlation value
  - (CountOutput, CountLine), (CountOutput, Cyclomatic), (CountLine, Cyclomatic), (CountLine, MaxNesting), (Cyclomatic, MaxNesting)
  - So we report the results of fanin and countline.

## How many technical debt can we map between a metrics file and Everton's summary file?

\*\_v1 means the version that introduces technical debt and \*\_v2 means the last version that technical debt was found.

```
# the number of technical debt that cannot be mapped between a metrics file  
#and Everton's summary file  
tapply(data$version_name, fc, function(x){sum(x == -1) })
```

```
##      apache-ant apache-jmeter      jruby  
##           3           0           4
```

```
tapply(data$CountInput_v1, fc, function(x){sum(x == -1) })
```

```
##      apache-ant apache-jmeter      jruby  
##           17           9          114
```

```
tapply(data$CountInput_v2, fc, function(x){sum(x == -1) })
```

```
##      apache-ant apache-jmeter      jruby  
##           4           3           82
```

```
# the number of technical debt that have metrics in both versions  
# of introduction and last_found  
a <- data[(data[, "CountInput_v1"] != -1 & data[, "CountInput_v2"] != -1), ]  
fc.a <- factor(a$Project)  
tapply(a$version_name, fc.a, length)
```

```
##      apache-ant apache-jmeter      jruby  
##           67          169          268
```

### Observation

- jruby misses 114 technical debt in v1 and 82 in v2.
- 67 (ant), 169(jmeter) and 268(jruby) technical debt has metrics in both versions of introduction and last\_found.
- We need to discuss how to solve such missed technical debt.

## How much is the interest?

We target 67 (ant), 169(jmeter) and 268(jruby) technical debt in this analysis. For each technical debt, we measure the ratio of metrics value of v2 in it of v1. We use 5 metrics as interest.

### CountInput (fanin)

```
# interest of CountInput (fanin)
idx <- a[,"CountInput_v1"] == 0
sum(idx)
```

```
## [1] 17
```

```
b <- a[!idx, ]
fc.b <- factor(b$Project)
interest <- (b[,"CountInput_v2"] ) / (b[,"CountInput_v1"])

# summary of interest for all technical debt
tapply(interest, fc.b, summary)
```

```
## $`apache-ant`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4615  1.0000  1.0000  1.0520  1.0190  2.5000
##
## $`apache-jmeter`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.900  1.000  1.000  1.041  1.000  2.000
##
## $jruby
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3125  1.0000  1.0000  1.1610  1.0830  9.0000
```

```
# the number of the percentage of positive interest, same interest, positive interest, negative interest
tapply(interest, fc.b, function(x){c( round((sum(x > 1)/ length(x) * 100)), sum(x==1), sum(x > 1), sum(x < 1))})
```

```
## $`apache-ant`
## [1] 25 36 16 12
##
## $`apache-jmeter`
## [1] 22 122 35 2
##
## $jruby
## [1] 27 153 70 41
```

```
# summary of interest for only technical debt that has positive / negative value.
tapply(interest, fc.b, function(x){summary(subset(x, x !=1))})
```

```
## $`apache-ant`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4615  0.7500  1.0830  1.1200  1.2710  2.5000
##
## $`apache-jmeter`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.900  1.083  1.133  1.178  1.222  2.000
##
## $jruby
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3125  0.8167  1.1670  1.3820  1.5000  9.0000
```

## CountLine

```
# interest of CountLine (LOC)
idx <- a[,"CountLine_v1"] == 0
sum(idx)
```

```
## [1] 0
```

```
b <- a[!idx, ]
fc.b <- factor(b$Project)
interest <- (b[,"CountLine_v2"] ) / (b[,"CountLine_v1"])
```

```
# summary of interest for all technical debt
tapply(interest, fc.b, summary)
```

```
## $`apache-ant`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1500  1.0000  1.0000  0.9944  1.0600  1.5710
##
## $`apache-jmeter`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.2951  1.0000  1.0000  1.0260  1.0000  2.1940
##
## $jruby
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03409 1.00000 1.00000 1.01800 1.04100 4.62500
```

```
# the number of the percenage of positive interest, same interest, positive interest, negative interest
tapply(interest, fc.b, function(x){c( round((sum(x > 1)/ length(x) * 100)), sum(x==1), sum(x > 1), sum(x < 1))})
```

```
## $`apache-ant`
## [1] 36 30 24 13
##
## $`apache-jmeter`
## [1] 22 120 38 11
##
## $jruby
## [1] 28 130 74 64
```

```
# summary of interest for only technical debt that has positive / negative value.
tapply(interest, fc.b, function(x){summary(subset(x, x !=1))})
```

```
## $`apache-ant`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.1500  0.8977  1.0530  0.9899  1.1050  1.5710
##
## $`apache-jmeter`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.2951  1.0110  1.0570  1.0910  1.1560  2.1940
##
## $jruby
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03409 0.83330 1.03500 1.03400 1.20000 4.62500
```

## Observation

- 22%-36% of technical debt has positive interest.
- If we focus on technical debt that has more than 1 interest
  - the number of technical debt that have positive interest is more than negative interest one

```
# the top interest tech debt?
met_v1 <- "CountInput_v1"
met_v2 <- "CountInput_v2"

data.CountInput <- data[,c("Project", "Method_Signature", met_v1, met_v2)]
data.CountInput <- data.CountInput[(data.CountInput[, met_v1] != -1 & data.CountInput[, met_v2] != -1),
idx <- order(data.CountInput[,met_v2] - data.CountInput[,met_v1], decreasing = T)
head(data.CountInput[idx,])
```

```
##      Project
## 797   jruby
## 805   jruby
## 419   jruby
## 227   jruby
## 816   jruby
## 649   jruby
##
##                                     Method_Signature
## 797                                     getObject()
## 805                                newIOException(java.io.IOException)
## 419                                     inspect(org.jruby.ast.Node)
## 227                                     inspect(org.jruby.ast.Node)
## 816  format(java.util.Date, java.lang.StringBuffer, java.text.FieldPosition)
## 649  unpack(org.jruby.Ruby, org.jruby.util.ByteList, org.jruby.util.ByteList)
##      CountInput_v1 CountInput_v2
## 797             19             65
## 805             25             65
## 419            113            144
## 227            114            142
## 816             26             48
## 649             10             21
```

## Observation

- [Discuss] if technical debt has same version and same method signature, should we remove one of them?
  - We solved the above point.

[Emad] I think one thing to measure is the metric value in v-1/metric value in v-2. Of course we can only do this for non-zero differences.

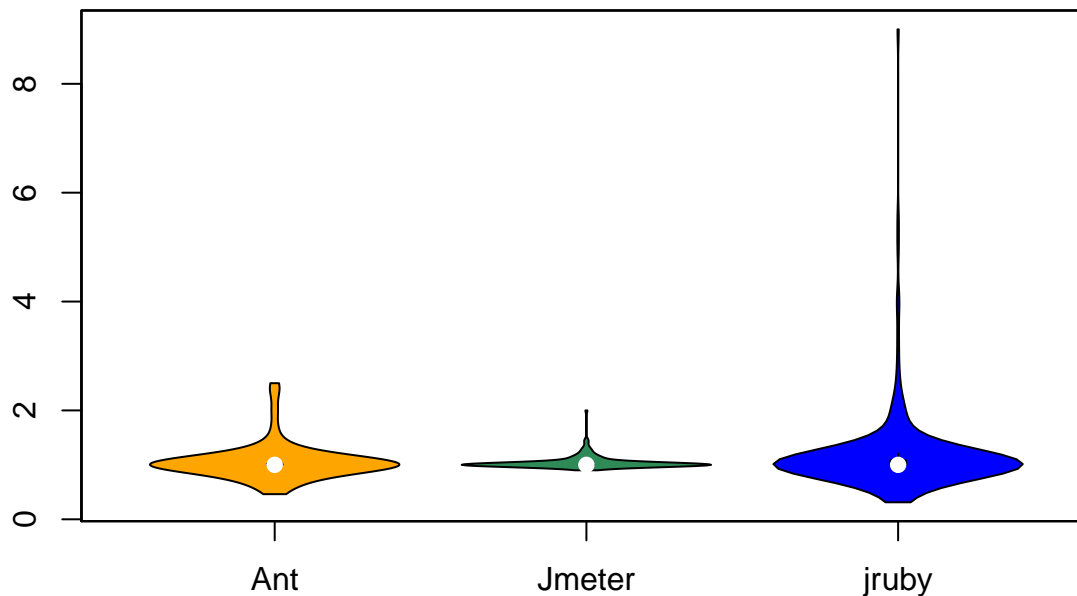
```
data.CountInput = subset(data.CountInput, data.CountInput$CountInput_v1 !=0)

library(vioplot)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
idx.ant <- data.CountInput[, "Project"] == "apache-ant"  
idx.jmeter <- data.CountInput[, "Project"] == "apache-jmeter"  
idx.jruby <- data.CountInput[, "Project"] == "jruby"  
  
ant <- data.CountInput[idx.ant, met_v2] / data.CountInput[idx.ant, met_v1]  
jmeter <- data.CountInput[idx.jmeter, met_v2] / data.CountInput[idx.jmeter, met_v1]  
jruby <- data.CountInput[idx.jruby, met_v2] / data.CountInput[idx.jruby, met_v1]  
  
plot(0, 0, type = "n", xlab = "", ylab = "", axes = FALSE,  
     xlim = c(0.5, 3.5), ylim = range(c(ant, jmeter, jruby)))  
  
axis(side = 1, at = 1:3, labels = c("Ant", "Jmeter", "jruby"))  
axis(side = 2)  
  
vioplot(ant, at = 1, col = "orange", add = TRUE)  
vioplot(jmeter, at = 2, col = "seagreen", add = TRUE)  
vioplot(jruby, at = 3, col = "blue", add = TRUE)
```



```
# How much percentage does technical debt has double interest?  
sum(ant >= 2)
```

```
## [1] 3
```

```
sum(jmeter >= 2)
```

```
## [1] 1
```

```
sum(jruby >= 2)
```

```
## [1] 17
```

### Observation

- There are several technical debt of which the ratio is more than 2. This means that the dependency of technical debt becomes double before being removed.

## period

### CountInput (fanin)

```
# interest of CountInput (fanin)
idx <- a[, "CountInput_v1"] == 0
b <- a[!idx, ]
interest <- ((b[, "CountInput_v2"] ) - (b[, "CountInput_v1"])) / (b[, "CountInput_v1"]) * 100
year <- b[, "year"]

summary(as.double(year))
```

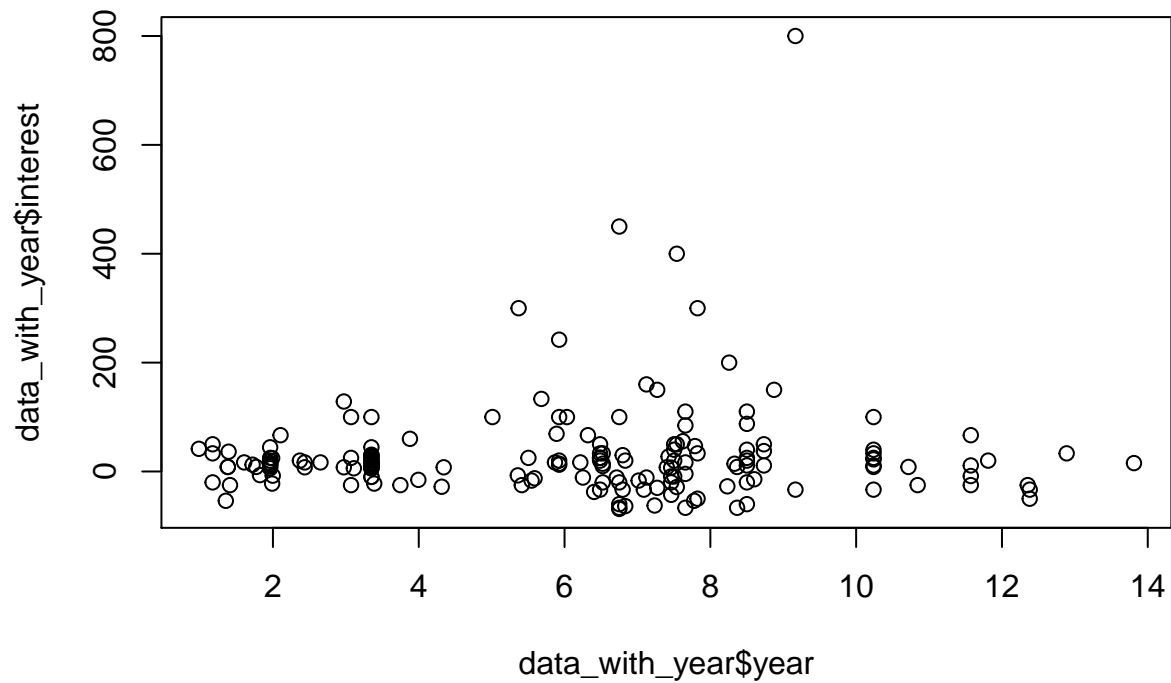
```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.00274  2.99900  5.86800  5.19700  7.17800 13.95000
```

```
# we focus on technical debt that has non 0 interest.
d <- interest != 0

data_with_year = data.frame(year=as.double(year[d]), interest=interest[d], project=b[d, "Project"])
fc.d <- factor(data_with_year$project)

plot(data_with_year$year, data_with_year$interest)
```





```
summary(data_with_year$interest/data_with_year$year)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -39.790 -2.179   2.963   5.720   7.523   87.270
```

```
tapply(data_with_year$interest/data_with_year$year, fc.d, summary)
```

```
## $`apache-ant`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -39.79000 -2.40200   0.88700  -0.08328   2.47800   23.47000
##
## $`apache-jmeter`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.984   3.158   5.969   6.986   8.049   29.840
##
## $jruby
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -17.060  -3.035   2.681   6.763   9.366   87.270
```

## Observation

- per year, around 0.8%-5.9% interest happens.