Linnéuniversitetet Kalmar Växjö

Software Quality (VT 2019)

Assignment 2: Quality Model / Indirect Metrics

Issued: 2019-05-06

To hand in by: 2019-05-20

In your second assignment, you are asked to design and implement an indirect metrics calculating a *Software Quality Model* using a set of given direct metrics as specified below. With their help you shall evaluate the *Maintainability* and the *Re-usability* of a Java project of your choice (preferably the one used in the Assignment 1).

1. Define top-level package metrics for each direct class metrics that counts the number of outlier classes in the packages. Use tree different methods to define the outliers:
    i. A class is an outlier if it has one of the highest (lowest) **15** direct metric values when low (high) values are actually good.
    ii. A class is an outlier if it has one of the highest (lowest) **15%** direct metric values when low (high) values are actually good.
    iii. A class is an outlier if it has a direct metric value within the top **15% of all values** measured.
2. Combine the package level metrics to indirect metrics for *Maintainability* and the *Re-usability* using the dependencies between these (package level) metrics and the factors and criteria as given in the Software Quality Matrix below. Each package should have 3 *Maintainability* and 3 *Re-usability* values (depending on the method: i, ii, iii).
3. Rank the top-level packages of your system wrt. *Maintainability* and the *Re-usability* (6 rankings). Compare the rankings of packages.
4. Prepare an Excel file (.xls, .xlsx) containing raw data as exported from VizzAnalyzer and the final measurement results. Moreover, you should prepare a short report (.doc, .docx, .pdf) describing how you derived the final measurement results from the raw data and your evaluation of the comparison of rankings.
5. Submit your results by email to rudiger.lincke@lnu.se, using 4DV607-Assignment 2 as subject:
    (1) Attach a .zip archive containing your .pdf (.doc, .docx) report and .xls (.xlsx) raw data.
    (2) Do not include the source code of the analyzed project! Do not use any other archive format than .zip.

Hints:

– `VizzMaintenance` implements the direct metrics. Outlier values (as defined above using iii) are marked in red in `VizzMaintenance` table view – ignore this. As you know, `VizzMaintenance` also allows to copy the table and to drop it into Excel. So you can run `VizzMaintenance` and the direct metrics on the example project and export the result to Excel as in Assignment 1.

- Using Excel's filtering and statistics functions, you derive the package level and indirect metrics for the systems – you implement so to say the metrics in Excel.
- A short remainder on the Software Quality Model: the Software Quality Matrix below describes the relationship between the different metrics and the criteria that contribute to the factors *Maintainability* and the *Re-usability* – **ignore CDBC**.

  It can be read the matrix as follows:
  - no relation gray (blank)
  - direct coupling between metrics and criteria: + (plus sign)
  - highly direct coupling between metrics and criteria ++ (double plus sign)
  - inverse coupling between metrics and criteria – (minus sign)
  - highly inversely related – – (double minus sign)

  When aggregating the single package metric values, a single + or – is having a weight of 1, and a double + or – is having a weight of 2. The weight for combining the criteria to the factors is always 1.

Software Quality Matrix

		Main property	Re-Usability					Maintainability				
		Sub Property	Understandability for Reuse	Learnability for Reuse	Operability for Reuse - Programmability	Attractiveness for Reuse	Re-Usability Compliance	Analyzability	Changeability	Stability	Testability	Maintainability Compliance
Category	Sub-Category	Metric										
Complexity	size	LOC	--	-	-	++		--	--	-	--	
	structural C.	WMC	--	--	--	++		--	--	-	--	
Architecture & Structure	Inheritance	DIT	--	--	--	++		--	--	-	--	
		NOC	++	++	++	++		-	--	-	-	
	Coupling	CDBC	--	-	-	--		--	--	--	--	
		DAC	--	-	-	--		--	--	--	--	
	Cohesion	TCC	++	+	+	++		++	++	++	++	
Design	Documentatic	LOD	--	--	--	-		--	--	-	--	