An Empirical Comparison and Characterization of High Defect and High Complexity Modules*

A. Güneş Koru and Jeff Tian

November 19, 2002

Abstract

We observed that the most complex modules might have an acceptable quality and high defect modules are not necessarily the most complex ones. The clusters of modules with the highest defects are usually those whose complexity rankings are slightly below the most complex ones.

*This research is supported by NSF/CAREER Award CCR-9733588, THECB/ATP Award 003613-0030-1999 and Nortel Networks.

Motivation

- Prediction of problem-prone modules
  [Porter and Selby, 1990, Tian and Troster, 1998]:
  - historical data & expert estimation,
  - process and personnel characteristics,
  - internal product measures, such as complexity and size.
- Common observation & belief: Positive correlation between complexity and defect count.
- Common intuition: Positive correlation between complexity and number of failures
  [Munson and Khoshgoftaar, 1992].
- Different aspects of complexity behavior were expressed [Whittaker and Voas, 2000].
- Our study seeks empirical evidence related to these issues.

Motivation - II

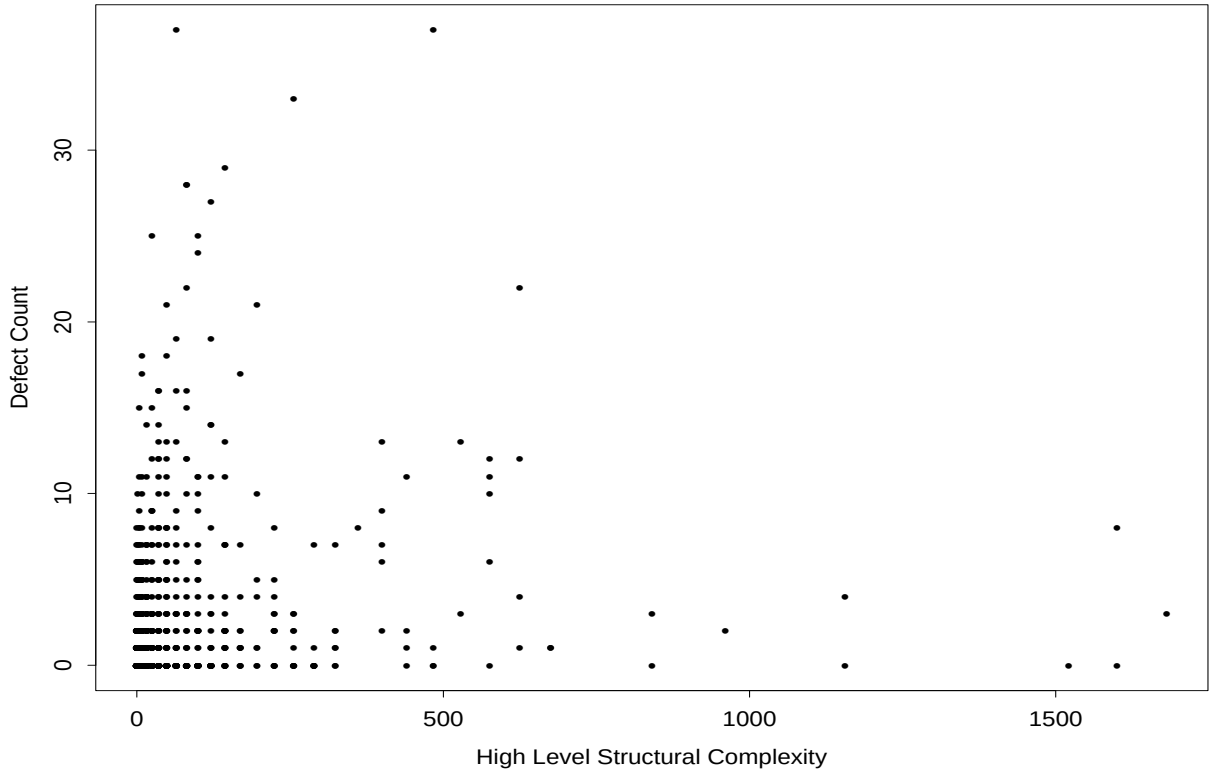

Figure 1: Data points

Motivation - III

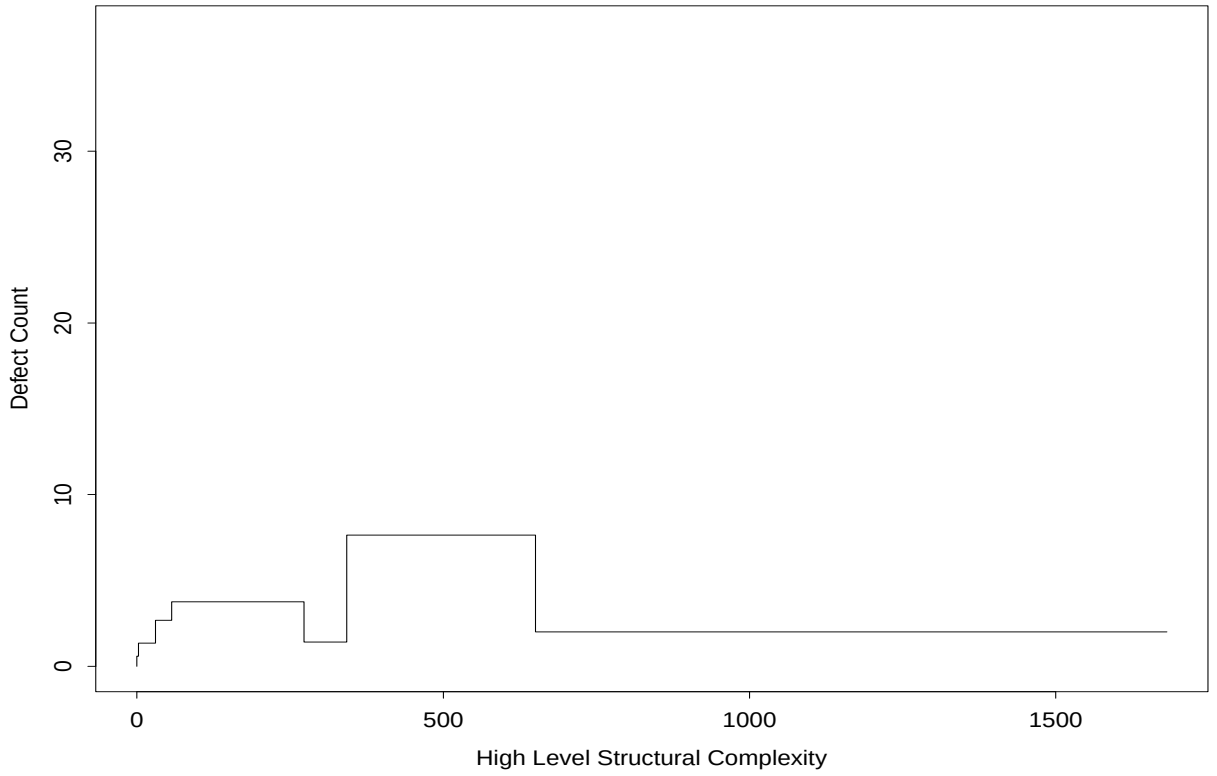

Figure 2: Piece-wise linear model showing means

Approach

- Analysis of complexity and defect data belong to large scale products.
- Locating high complexity (HC) and high defect (HD) module clusters.
  - Tree-based modeling.
- Characterization.
  - Piece-wise linear models.
- Statistical comparison to test sameness.
  - Hypothesis testing.
- Further analysis of top defect module clusters.
  - Complexity ranking.

Outline

1. Description of products and data.

2. Identification of HC and HD clusters:

   (a) Locating,
   (b) Characterization and its results.

3. Hypothesis Testing:

   (a) Purpose & Samples,
   (b) Test Statistic,
   (c) Hypotheses and Procedure,
   (d) Test results.

4. Complexity ranking of top defect clusters.

5. Conclusions.

1. Description of Products & Data

1. Products: Six large scale products are analyzed. Two from IBM and four from Nortel Networks. Size about one million LOC.
   IBM-LS: Relational database management system (RDMS). 1302 modules. Legacy system, written in PL/AS.
   IBM-NS: RDMS, 995 modules. New system, written in C/C++.
   NT1, NT2, NT3, NT4: Telecommunications software. 804, 1098, 712, and 900 modules respectively. In Protel.

2. Data: Available at module level.
   IBM-LS: 15 metrics of design, size, and, change.
   IBM-NS: 11 of the metrics used for IBM-LS.
   Nortel Products: 49 metrics of volume, testability, decision complexity, independent path, structuredness, dead code, readability and section dependability.

2. Identification of HC and HD clusters
(a) Locating - I

- Tree-Based Modeling
  - Variables:
    * Response: Defect count.
    * Predictor: One of the various metrics available.
  - Recursive binary partitioning of modules using certain cutoff values of the predictor variable.
  - Deviance reduction in defect count
  - Partitioning until a certain size or a deviance reduction threshold reached.
  - Average value of defect count is available at each node.

- HC cluster: The cluster with the highest values of the predictor variable (rightmost in a binary tree)

- HD cluster: The cluster with the highest average value of the defect count.

2. Identification of HC and HD clusters
(a) Locating - II

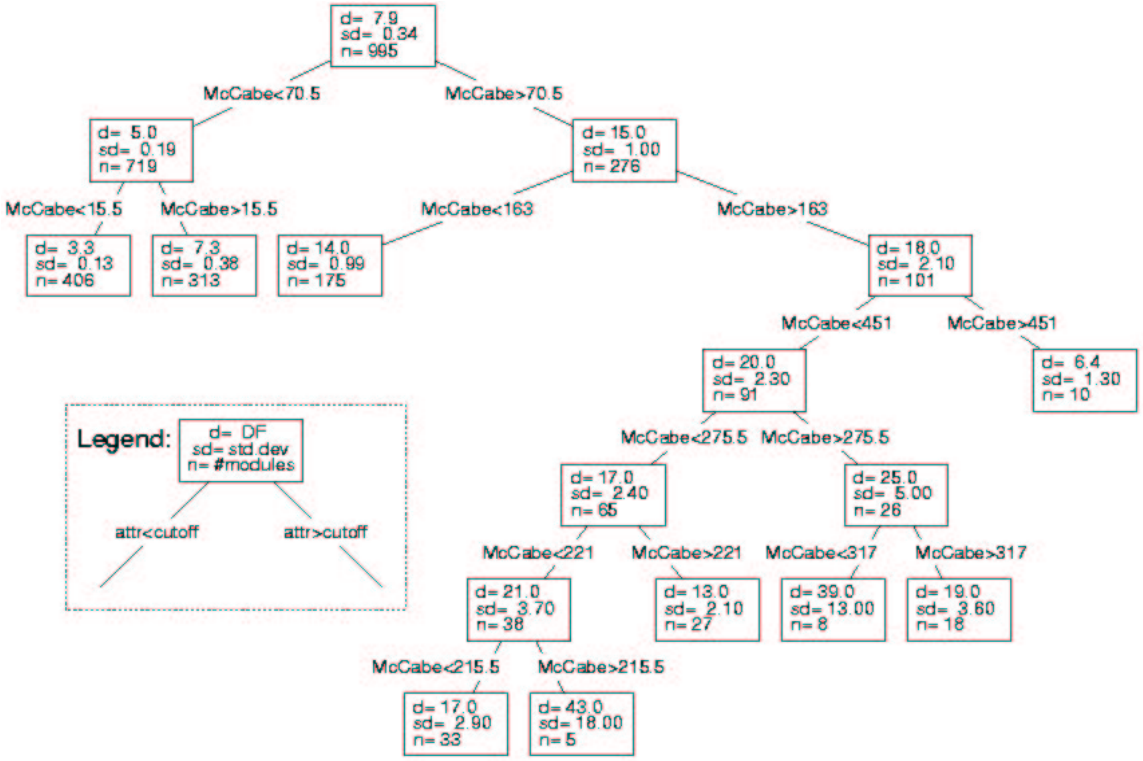

Figure 3: A sample tree-based model

2. Identification of HC and HD clusters (a) Locating - III

0. *Initialization.* Initialize a list, `Slist`, for the data sets to be partitioned, with the complete data set as the singleton element. Select the size and homogeneity thresholds T_s and T_h for the algorithm.

1. *Overall control.* Repeatedly remove a data set from `Slist` and execute step 2 until `Slist` becomes empty.

2. *Size test.* If $|S| < T_s$, stop; otherwise, execute steps 3 through 6. $|S|$ is the number of data points in set S.

3. *Defining binary partitions.* A binary partition divides S into two subsets using a *split condition* defined on a specific predictor p with a cutoff value c. Data points with $p < c$ form one subset (S_1) and those with $p \geq c$ form another subset (S_2).

4. *Computing predicted responses and prediction deviances.* The predicted response value $v(S)$ for a set S is the average over the set; i.e., $v(S) = \frac{1}{|S|}\sum_{i \in S}(u_i)$; and the prediction deviance is $D(S) = \sum_{i \in S}(u_i - v(S))^2$, where u_i is the response value for data point i.

5. *Selecting the optimal partition.* Among all the possible partitions (all predictors with all associated cutoffs), the one that minimizes the deviance of the partitioned subsets is selected; i.e., the partition with minimized $D(S_1) + D(S_2)$ is selected.

6. *Homogeneity test:* Stop if this partitioning cannot improve prediction accuracy beyond a threshold T_h, i.e., stop if $\left(1 - \frac{D(S1)+D(S2)}{D(S)}\right) \leq T_h$; otherwise, append S_1 and S_2 to *Slist*.

Figure 4: Algorithm for tree-based model construction

2. Identification of HC and HD clusters
(b) Characterization and Results - I

- Leaf nodes: Clusters with upper and lower bounds according to a predictor variable m.

- Piece-wise linear models.

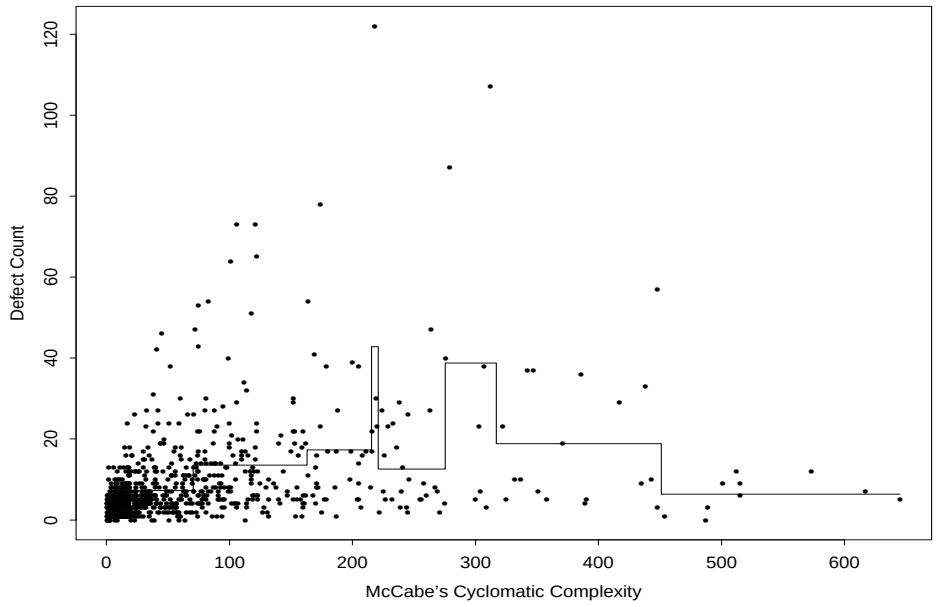

Figure 5: A sample linear piece-wise model.

2. Identification of HC and HD clusters
(b) Characterization and Results - II

- Models obtained using each available metric in every product.
- Common patterns observed.
- Categories:
  - $Type\ A$: HC and HD clusters are identical having the highest defect count.
  - $Type\ B$: HD cluster precedes the HC cluster.

  $Type\ B$ category further divided into two:

  - $Type\ B1$: HD cluster is the one immediately preceding the HC cluster.
  - $Type\ B2$: $Type\ B$ but not $Type\ B1$.

2.(b). Continued...

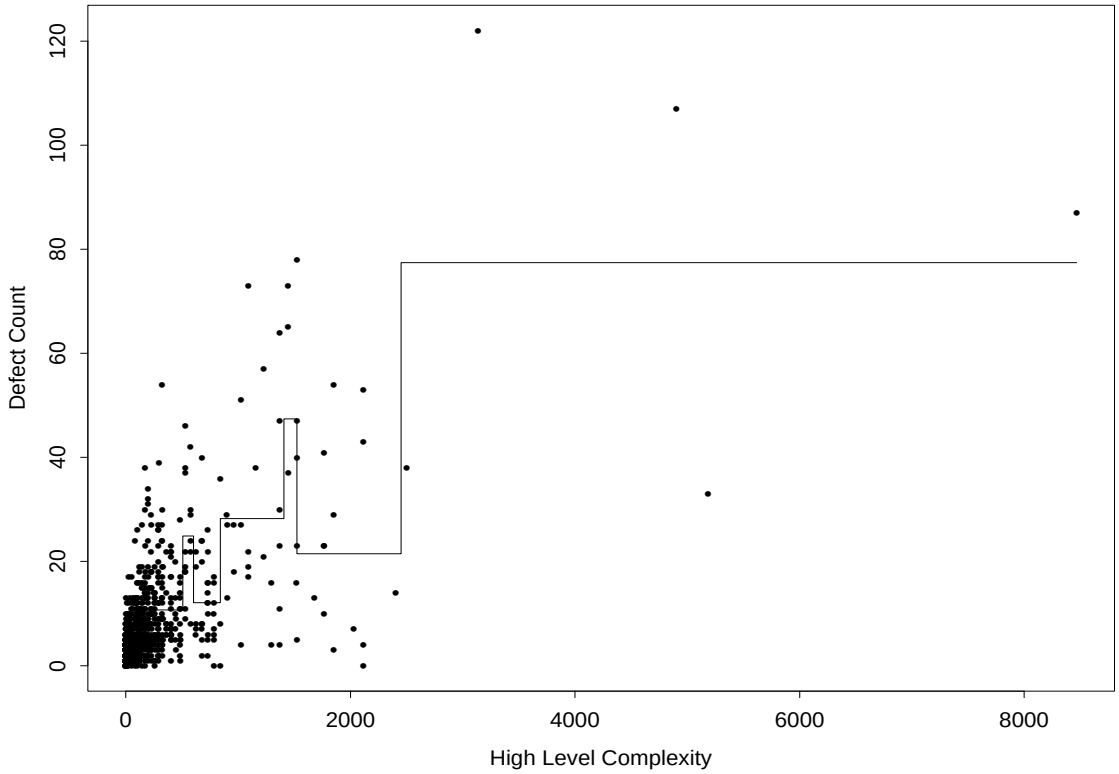

Figure 6: *Type A* example.

2.(b). Continued...

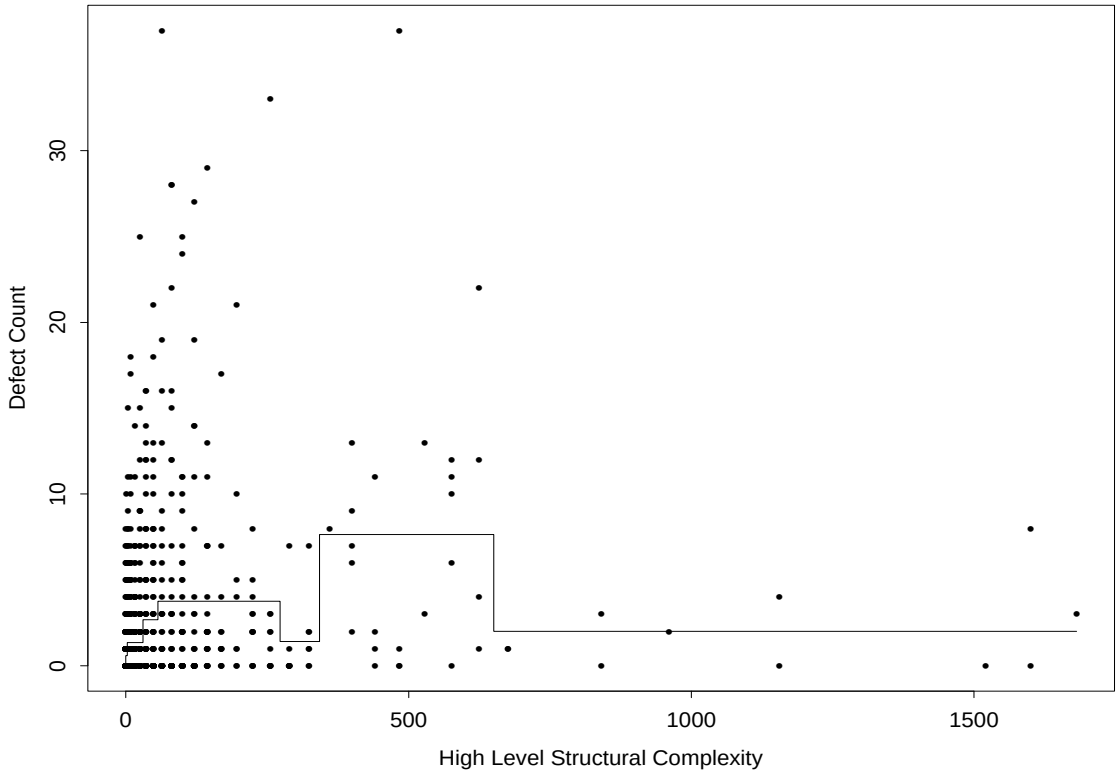

Figure 7: $Type\ B1$ example.

2.(b). Continued...

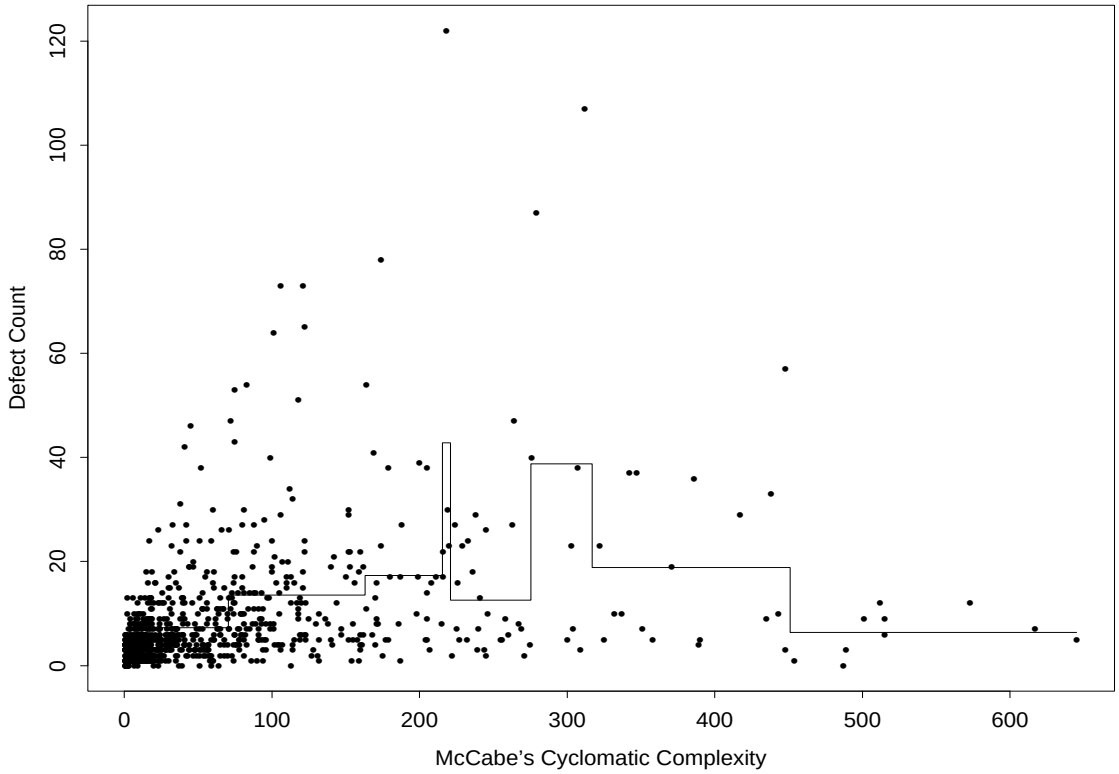

Figure 8: $Type\ B2$ example

2.(b). Continued...

Product Name	$Type\ A$	$Type\ B1$	$Type\ B2$
IBM-LS	0	5	10
IBM-NS	3	0	8
NT-1	16	6	27
NT-2	8	11	30
NT-3	7	18	24
NT-4	5	25	19
total	39 (17.57%)	65 (29.28%)	118 (53.15%)

Table 1: Characterization Results.

In all products, the number of $Type\ B$ models is much larger than the number of $Type\ A$ models.

3. Hypothesis Testing
(a) Purpose & Samples

- Aim: Making a statistical inference about the sameness of two samples - high complexity and high defect clusters.

- Sample Pairs
  - HC_p^O vs HD_p^O
  - HC_p^O vs $TopDF_p$
  - HC_p^U vs HD_p^U
  - HC_p^U vs $TopDF_p$.

$i =$	1	2	3	4	5	6
HC_i^O	474	1211	2945	6121	2921	2830
HC_i^U	257	611	620	857	656	878
HD_i^O	203	64	1083	1696	531	402
HD_i^U	139	43	501	483	228	117

Table 2: Number of data points.

3. Hypothesis Testing
(b) Test Statistic-I

- Student's t test could be used if samples had a normal distribution.

- The most suitable test statistic:

  Mann-Whitney U test (also known as Wilcoxon rank sum test)

3. Hypothesis Testing
(b) Test statistic - II

- Interested in ranks instead of raw measures
  - Shifts the focus to ordinal relationships (such as greater than, etc.)
  - Known properties are obtained

- Steps
  - Two samples are combined and sorted. Each observation gets a rank value.
  - For each sample, the sum of the ranks are calculated (R_1 and R_2)

- A U statistic is calculated as:

$$U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1 \qquad (1)$$

which is the difference between maximum possible sum of ranks and the observed sum of ranks.

3. Hypothesis Testing
(b) Test statistic - III

- U approximates to the normal distribution. $n_1(n_1+1)$ can be replaced with $n_2(n_2+1)$ and R_1 can be replaced with R_2.

- U_A and U_B are like mirror images and their sum is equal to $n_1 n_2$.

- The expected value of U:

$$\mu_U = \frac{n_1 n_2}{2} \tag{2}$$

- Variance

$$\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}} \tag{3}$$

- Standard score

$$z = \frac{U - \mu_U}{\sigma_U} \tag{4}$$

3. Hypothesis Testing
(b) Test statistic - IV
Example

- Sample A: 4.6, 5.1, 5.8, 6.5, 4.7, 5.2, 6.1, 7.2, 4.9, 5.5, 6.5
- Sample B: 5.2, 5.6, 6.8, 8.1, 5.3, 6.2, 7.7, 5.4, 6.3, 8.0
- Non-normal samples. n_A=11, $n_B = 10$.

raw measure	rank	from sample
4.6	1	A
4.7	2	A
4.9	3	A
5.1	4	A
5.2	5.5	A
5.2	5.5	B
5.3	7	B
5.4	8	B
5.5	9	A
5.6	10	B
5.8	11	A
6.1	12	A
6.2	13	B
6.3	14	B
6.5	15.5	A
6.5	15.5	A
6.8	17	B
7.2	18	A
7.7	19	B
8.0	20	B
8.1	21	B

Table 3: Sorted rank table.

Raw Measures		Ranked Measures		
Group A	Group B	Group A	Group B	
4.6	5.2	1	5.5	
4.7	5.3	2	7	
4.9	5.4	3	8	
5.1	5.6	4	10	
5.2	6.2	5.5	13	
5.5	6.3	9	14	
5.8	6.8	11	17	
6.1	7.7	12	19	
6.5	8.0	15.5	20	
6.5	8.1	15.5	21	A & B Combined
7.2		18		
sum of ranks		96.5	134.5	231
average of ranks		8.8	13.5	11

Table 4: Raw and ranked measures.

- The maximum ranks:
- MR_A: 11+12+13+14+15+16+17+18+19+20+21=176
- MR_B: 12+13+14+15+16+17+18+19+20+21=165
- Let us rewrite MR_A:
  - (11+0)+(11+1)+(11+2)+.....+(11+10) *or*
  - 11.10 + 0 + 1 + 2 + ... + 10 *or*
  - $n_1.n_2 + \frac{n_2(n_2+1)}{2}$

- That's why $U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$, the difference between the maximum rank-sum and the observed rank sum.

- In this example:
  - $U_A = 176 - 96.5 = 79.5$ and $U_B = 165 - 134.5 = 30.5$
  - $\mu_U = \frac{n_1 n_2}{2} = \frac{11.10}{2} = 55$
  - $\sigma_U = \sqrt{\frac{n_1 n_2 (n_1+n_2+1)}{12}} = \sqrt{\frac{11.10(11+10+1)}{12}} = 14.20$
  - $z = \frac{U-\mu_U}{\sigma_U} = \frac{79.5-55}{14.20} = 1.725$
  - Now the problem reduced to testing the null (default) hypothesis: H_0: $z = 0$ where H_A (alternative hypothesis) states the opposite.
  - Type I error: H_0 is rejected when it's really true. (α)
  - Type II error: H_0 is not rejected when it should be (when H_A is true)
  - α values of 0.05 and 0.01 correspond to 1.96 and 2.58 values of $|z_{critical}|$.
  - H_0 accepted.

3. Hypothesis Testing
(c) Hypotheses & Procedure

- In a generic form:
  - H_0: Two samples, S_1 and S_2, have been drawn from the same population, or equivalently from two different populations having the same mean.
  - H_A: The distribution for population S_1 is shifted to the left or to the right of that for S_2.

- Procedure:
  1. Produce the two samples subject to hypothesis testing.
  2. Apply the $Mann\text{-}Whitney\ U$ test statistic and obtain the z value of Formula 4.
  3. Decide upon a significance level, α. Then the critical value $|z_{critical}|$, is determined. α values of 0.05 and 0.01 correspond to 1.96 and 2.58 values of $|z_{critical}|$.
  4. Accept H_0 if $|z| \leq |z_{critical}|$, otherwise reject H_0 in favor of H_A.

3. Hypothesis Testing
(d) Results

Name	S_1	S_2	z
$H1_0$	HC_1^O	HD_1^O	−10.4786
$H2_0$	HC_1^U	HD_1^U	−6.3903
$H3_0$	HC_1^O	$Top25DF_1$	−8.2033
$H4_0$	HC_1^O	$Top100DF_1$	−12.9213
$H5_0$	HC_1^U	$Top25DF_1$	−7.9312
$H6_0$	HC_1^U	$Top100DF_1$	−11.9387
$H7_0$	HC_2^O	HD_2^O	−7.8070
$H8_0$	HC_2^U	HD_2^U	−4.9644
$H9_0$	HC_2^O	$Top25DF_2$	−8.1396
$H10_0$	HC_2^O	$Top100DF_2$	−13.9276
$H11_0$	HC_2^U	$Top25DF_2$	−8.2194
$H12_0$	HC_2^U	$Top100DF_2$	−14.1263
$H13_0$	HC_3^O	HD_3^O	−5.4202
$H14_0$	HC_3^U	HD_3^U	−2.7851
$H15_0$	HC_3^O	$Top25DF_3$	−7.8279
$H16_0$	HC_3^O	$Top100DF_3$	−12.3365
$H17_0$	HC_3^U	$Top25DF_3$	−8.3677
$H18_0$	HC_3^U	$Top100DF_3$	−14.0605
$H19_0$	HC_4^O	HD_4^O	−10.7930
$H20_0$	HC_4^U	HD_4^U	−7.0602
$H21_0$	HC_4^O	$Top25DF_4$	−8.0769
$H22_0$	HC_4^O	$Top100DF_4$	−13.7444
$H23_0$	HC_4^U	$Top25DF_4$	−8.3571
$H24_0$	HC_4^U	$Top100DF_4$	−14.7316
$H25_0$	HC_5^O	HD_5^O	−15.3403
$H26_0$	HC_5^U	HD_5^U	−9.6969
$H27_0$	HC_5^O	$Top25DF_5$	−7.9758
$H28_0$	HC_5^O	$Top100DF_5$	−12.7414
$H29_0$	HC_5^U	$Top25DF_5$	−8.3441
$H30_0$	HC_5^U	$Top100DF_5$	−13.9564
$H31_0$	HC_6^O	HD_6^O	−21.6221
$H32_0$	HC_6^U	HD_6^U	−8.8462
$H33_0$	HC_6^O	$Top25DF_6$	−8.3342
$H34_0$	HC_6^O	$Top100DF_6$	−14.3073
$H35_0$	HC_6^U	$Top25DF_6$	−8.5951
$H36_0$	HC_6^U	$Top100DF_6$	−15.0384

Table 5: Hypothesis testing results.

All null hypotheses are rejected in favor of their corresponding alternative hypotheses.

4. Complexity Ranking of HD and Top DF clusters

- We added a rank column in our "original" data sheet, next to each column of complexity data.

- Then we obtained the high-defect and top-defect clusters and examine their complexity range.

Cluster Name	Cluster Size	Min. Rank	Max. Rank	Possible Rank Range	Avg. Rank	Avg. Rank Percentile
$Top2percent_1$	26	413.5	1295	1–1302	1107.4	85.51%
$Top5percent_1$	65	128.5	1295	1–1302	1081.1	83.48%
$Top2percent_2$	20	53	995	1–995	800	80.40%
$Top5percent_2$	50	53	995	1–995	786	78.99%
$Top2percent_3$	16	2	804	1–804	564.5	70.21%
$Top5percent_3$	40	2	804	1–804	552.8	68.76%
$Top2percent_4$	22	2	1098	1–1098	786.7	71.65%
$Top5percent_4$	55	2	1098	1–1098	735.3	66.97%
$Top2percent_5$	14	2	711	1–712	494.5	69.55%
$Top5percent_5$	36	1	711	1–712	468.8	65.94%
$Top2percent_6$	18	1	900	1–900	730.6	81.18%
$Top5percent_6$	45	1	900	1–900	685.2	76.13%

Table 6: High and top complexity ranking.

- Fairly high in complexity but not the highest.

5. Conclusions

- There is usually a correlation between complexity and defect count however HD cluster is usually not the most complex cluster.

- This point was also observed by other researchers.

- The high defect modules are typically those measured at fairly high percentile on various complexity scales, but not the highest.

  – skill, effort, time, problem complexity.

- Possibility of some *worst* complexity, those "not too big (complex) not too small (simple)", which might contain the highest number of defects.

  – Similar to an optimum size.
  – Particular attention to modules whose measured complexity falls slightly below the most complex ones.

References

[Munson and Khoshgoftaar, 1992] Munson, J. C. and Khoshgoftaar, T. M. (1992). The detection of fault-prone programs. *IEEE Trans. on Software Engineering*, 18(5):423–433.

[Porter and Selby, 1990] Porter, A. A. and Selby, R. W. (1990). Empirically guided software development using metric-based classification trees. *IEEE Software*, 7(2):46–54.

[Tian and Troster, 1998] Tian, J. and Troster, J. (1998). A comparison of measurement and defect characteristics of new and legacy software systems. *Journal of Systems and Software*, 44(2):135–146.

[Whittaker and Voas, 2000] Whittaker, J. A. and Voas, J. (2000). Toward a more reliable theory of software reliability. *IEEE Computer*, 33(12):36–42.