Software Quality Management

Software Quality Analysis

Lecturer: **Nguyễn Ngọc Tú**
Email: *Tu.NguyenNgoc@hoasen.edu.vn*
Web: *sites.google.com/site/QuanLyChatLuongPhanMem/*
Face Group: *www.facebook.com/groups/SoftwareQualityManagement/*
#AdTekDev #ICoTek #VNASQ #VNSQA #VNSoftwareTesting

Outline

- Defect Analyses
- ODC
- Techniques
  - Qualitative
  - Statistical
  - Model: PCA, ANN, TBM, …

Defect Analyses

- A defect is discovered,
  - various individual analyses can be performed.
- defect data are accumulated over time
  - collective analyses can be performed
- General defect analyses:
  - Questions: what/where/when/how/why?
  - Distribution/trend/causal analyses …

Defect Analyses – distribution

Type	Description	#failures
A	permission denied	2079
B	no such file or directory	14
C	stale NFS file handle	4
D	client denied by server configuration	2
E	file does not exist	28,631
F	invalid method in request	0
G	invalid URL in request connection	1
H	mod_mime_magic	1
I	request failed	1
J	script not found or unable to start	27
K	connection reset by peer	0
All types		**30,760**

Defect Analyses – trend analysis

Injection phase	Removal Phase						
	req.	*spec.*	*design*	*coding*	*testing*	*post-rel*	*all phases*
requirement	10	22	8	0	5	2	47
specification		10	20	2	0	1	33
design			52	120	32	5	209
coding				198	320	46	564
testing					58	7	65
post-release						2	2
all phases	10	32	80	320	415	63	920

DRE

$$DRE = \frac{Removed\ pre\_release\ Defect}{pre\_release + post\_release}$$

ODC

- Orthogonal Defect Classification, or ODC, developed initially at IBM (Chillarege et al., 1992)
  - is the most influential among general frameworks for software defect classification and analysis.
  - to identify problematic areas, and to improve overall software product quality

ODC

- Key elements of ODC
  - Aim: tracking/analysis/improve
  - Approach: classification and analysis
  - Key attributes of defects
  - Views: both failure and fault
  - Applicability: inspection and testing
  - Analysis: attribute focusing
  - Need for historical data

ODC

- 8 attributes
  - Activity
    - refers to the actual process step (code inspection, function test, etc.) when defects are discovered.
  - Trigger
    - describes the environment or condition that had to exist to expose the defects.
  - Impact
    - refers to either perceived or real impact on users.
  - Target
    - represents the high-level identity (i.e., design, code, ID, etc.) of the entity that was fixed.

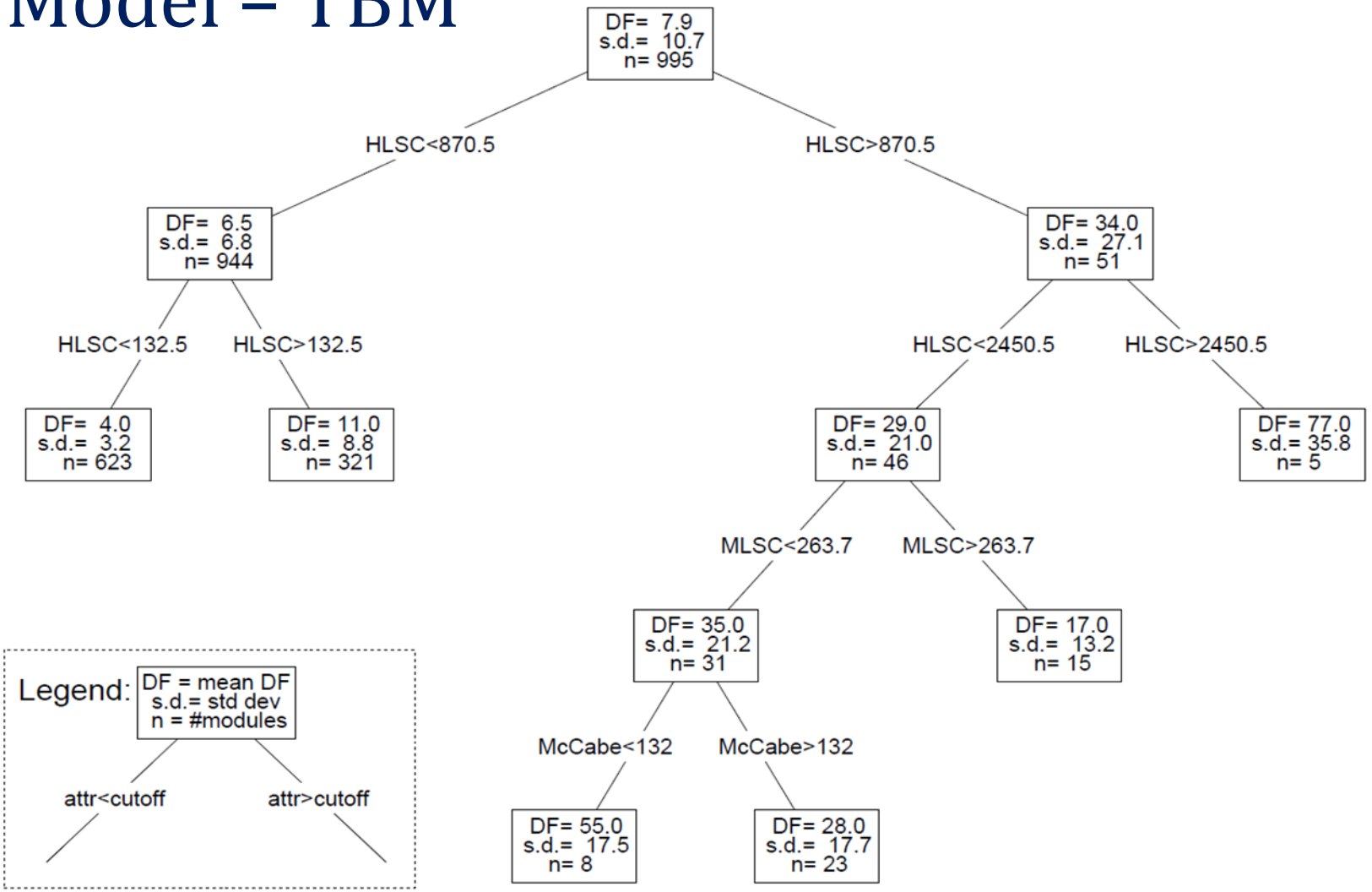
ODC

- 8 attributes
  - Type
    - represents the nature of the actual correction that was made.
  - Qualifier
    - specifies whether the fix that was made was due to missing, incorrect, or extraneous code or information.
  - Source
    - indicates whether the defect was found in code written in-house, reused from a library, ported from one platform to another, or outsourced to a vendor.
  - Age
    - identifies the history of the target (i.e., design, code, ID, etc.) that had the defect.

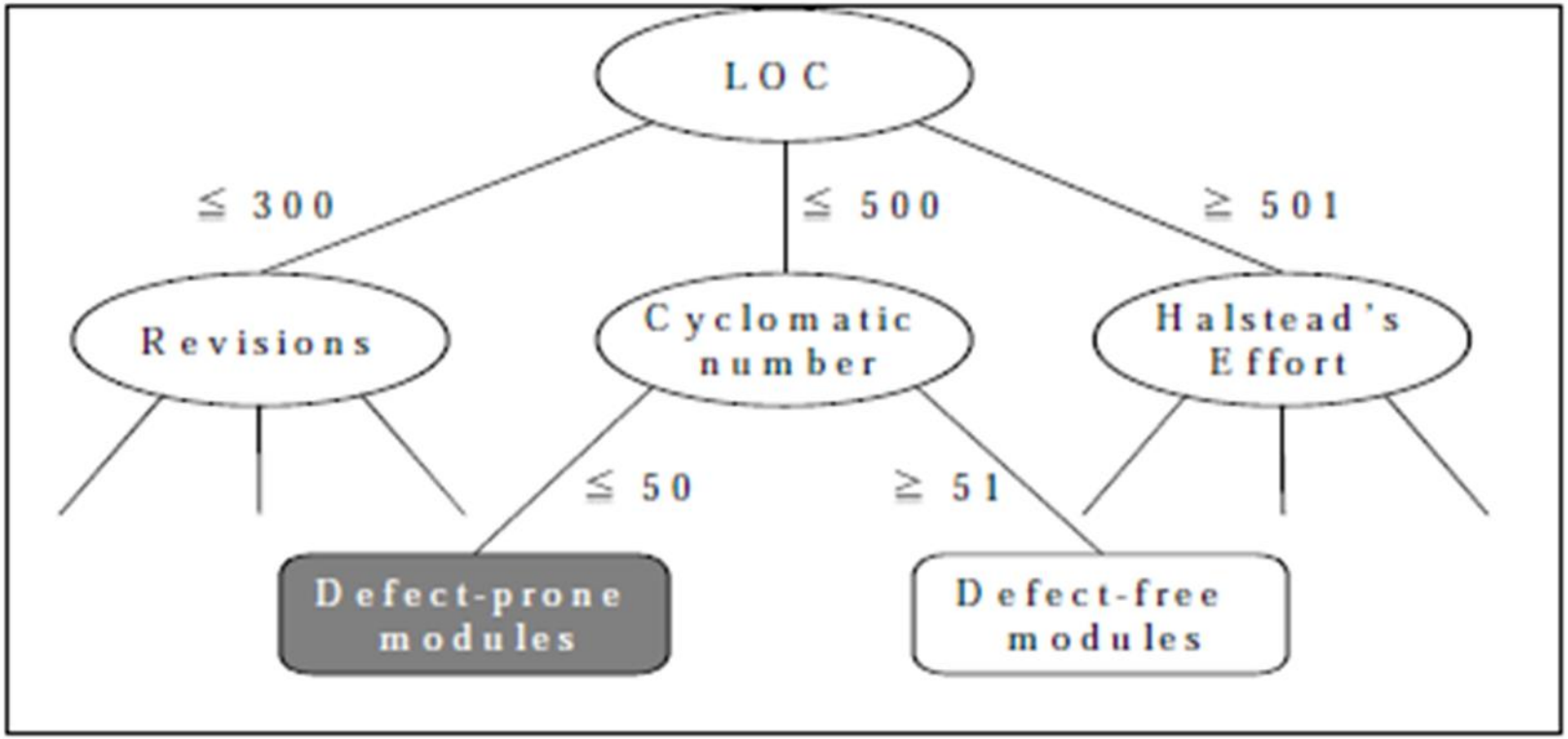
Model – TBM

0. *Initialization.* Set the list, `Slist`, to contain only the complete data set as its 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. For numerical p, it can be defined 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). If p is a categorical variable, a binary partition is a unique grouping of all its category values into two mutually exclusive subsets S_1 and 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; that is, $v(S) = \frac{1}{|S|}\sum_{i \in S}(v_i)$; and the prediction deviance is $D(S) = \sum_{i \in S}(v_i - v(S))^2$, where v_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 or binary groupings), the one that minimizes the deviance of the partitioned subsets is selected; that is, the partition with minimized $D(S_1) + D(S_2)$ is selected.
6. *Homogeneity test:* Stop if $\left(1 - \frac{D(S_1)+D(S_2)}{D(S)}\right) \leq T_h$ (that is, stop if there is no substantial gain in prediction accuracy in further partitioning); otherwise, append S_1 and S_2 to $Slist$.

Model – TBM


DF= 7.9
s.d.= 10.7
n= 995
HLSC<870.5
HLSC>870.5
DF= 6.5
s.d.= 6.8
n= 944
DF= 34.0
s.d.= 27.1
n= 51
HLSC<132.5
HLSC>132.5
HLSC<2450.5
HLSC>2450.5
DF= 4.0
s.d.= 3.2
n= 623
DF= 11.0
s.d.= 8.8
n= 321
DF= 29.0
s.d.= 21.0
n= 46
DF= 77.0
s.d.= 35.8
n= 5
MLSC<263.7
MLSC>263.7
DF= 35.0
s.d.= 21.2
n= 31
DF= 17.0
s.d.= 13.2
n= 15
McCabe<132
McCabe>132
DF= 55.0
s.d.= 17.5
n= 8
DF= 28.0
s.d.= 17.7
n= 23
Legend:
DF = mean DF
s.d.= std dev
n = #modules
attr<cutoff
attr>cutoff

Model – TBM


LOC
≤ 300
≤ 500
≥ 501
Revisions
Cyclomatic number
Halstead's Effort
≤ 50
≥ 51
Defect-prone modules
Defect-free modules

Q/A ?!