Predicting Fault Incidence Using Software Change History

Todd L. Graves, Alan F. Karr, J.S. Marron, and Harvey Siy

Presented by Scott McMaster
CMSC838M – Fall 2002
University of Maryland

Code Decay

- Software structure degrades over time
  - Why?
- Changes can become:
  - Costly
  - Time-consuming
  - ***Fault-producing***
    - When one fix leads to one fault on average, what's the use? We should just go home...

Fault Analysis

- Usually looks at:
  - Number of faults remaining
  - Explaining the number of faults found
- *This paper assumes that new faults are added as the system is changed.*

Definitions

- *Module*
  - Collection of related files
- *Delta*
  - Change to a module
- *Age*
  - Weighted average of dates of deltas weighted with sizes of the deltas

Predictors of the Number of Faults

- Product Measures
  - Computed from syntactic data
- Process Measures
  - Computed from change and defect histories

Product Measures

- Lines of Code
- Other Complexity Measures (McCabe, etc.)
  - Highly correlated with lines of code
- Not very good predictors of faults

Process Measures

- Number of past faults
  - "Stable model"
- Number of historical deltas to a module
- Average age of the code
- Development organization

Process Measures Continued

- Number of developers making changes
- Module's connection to other modules
  - In terms of the modules being changed together
- "Weighted time damp model"
  - More recent changes contribute more to fault potential

The Experiment

- 1.5 million LOC legacy from a telephone switching system
- Looked at data from a two-year period
- Modules have different versions (domestic, international, and common)

IMRs

- "Initial Modification Request"
  - Read: "Change Request"
  - Official record of a problem to be solved
  - Two types, set by originator
    - "Bug" – bug fix or request for missing feature
    - "New" – new feature
  - Typically results in several deltas

Data Sources

- Data sources:
  - IMR database
    - Only examine those classified as bug fixes
  - Delta database
    - Read "Change Management"
    - Deltas associated with IMRs
  - Source code
    - Comments included in LOC counts

Models

- Hypothesized formulas for fault prediction
- Composed of one or more variables (such as deltas, age, or lines of code)
- Different models are postulated and their fault-predicting powers are statistically examined

Statistics Technique

- Generalized Linear Models (GLMs)
  - Curve-fitting technique (i.e. attacks the same type of problem as linear regression / least-squares)
  - Effective on Poisson distributions
  - Made a logarithmic function of the mean to be linear in the variables
  - Error measure chosen to minimize the effects of having radically different sizes and fault counts of modules
    - Deviance function for the Poisson distribution

Simulations

- Used to compute the significance of variables in models
- Generated synthetic fault data and compared deviances between models

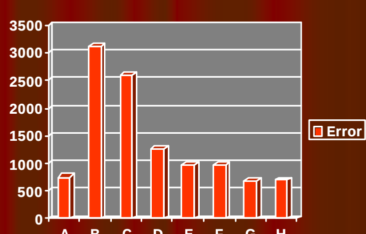
Basic Generalized Linear Models

- Stable Model
  - Assumes that fault generation dynamics for a module remain stable over time
    - In other words, if you found 100 faults last year, you'll find 100 this year
  - Insight-free
  - Implicitly incorporates many of the other predicting variables
- Null Model
  - All modules have the same number of faults
- Organization Only
  - Prediction by module version (international, domestic, or common)

Results

	Model	Error
A	Stable	757.4
B	Null	3108.8
C	Organization Only	2587.7
D	0.84 log (lines/1000)	1271.4
E	0.14 log (lines/100) + 1.19 log (deltas/1000)	980.0
F	1.05 log (deltas/1000)	985.1
G	0.07 log (lines/1000) + 0.95 log (deltas/1000) - 0.44 age	696.3
H	1.02 log (deltas/1000) – 0.44 age	697.4

Results Again



Observations

- Predictors
  - Deltas are a better measure of fault likelihood than lines
  - Age idea is helpful to incorporate, too
- Non-predictors
  - Lines don't help much
    - Complexity metrics were predicable from lines of code
  - Number of developers working on the code
  - Module's connectivity to other modules

Correlation of Complexity Metrics

	1	2	3	4	5	6	7	8	9	10	11	12
1 Lines Of Code	1	.97	.88	.88	.91	.99	.98	.92	.97	.83	.72	.35
2 McCabe V(G)1	.97	1	.88	.90	.88	.95	.95	.89	.93	.86	.76	.39
3 Functions	.88	.88	1	.82	.89	.85	.84	.91	.84	.78	.65	.39
4 Breaks	.88	.90	.82	1	.83	.86	.85	.85	.85	.78	.67	.27
5 Unique Operators	.91	.88	.89	.83	1	.89	.87	1.00	.94	.46	.47	.48
6 Total Operands	.99	.95	.85	.86	.89	1	1.00	.90	.98	.83	.72	.31
7 Program Volume	.98	.95	.84	.85	.87	1.00	1	.88	.97	.87	.74	.28
8 Expected Length	.92	.89	.91	.85	1.00	.90	.88	1	.94	.69	.53	.42
9 Variable Count	.97	.93	.84	.86	.94	.98	.97	.94	1	.77	.80	.38
10 MaxSpan	.83	.86	.78	.78	.45	.85	.87	.69	.77	1	.92	-0.10
11 MeanSpan	.72	.76	.65	.67	.47	.72	.74	.53	.80	.92	1	-0.15
12 Prog Level	.35	.39	.39	.27	.48	.31	.28	.42	.38	-0.10	-0.15	1

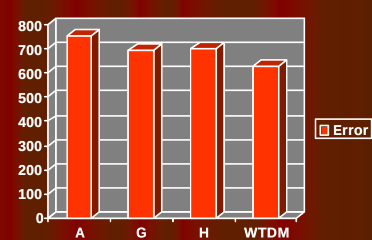
Weighted Time Damp Model

- Considers the fault potential to be a weighted sum of all historical changes in a module
- Contribution of a change goes down about 50% per year
- Assumes that old changes have been fixed or proven to be fault-free
- Treats changes individually

Results

	Model	Error
A	Stable	757.4
B	Null	3108.8
C	Organization Only	2587.7
D	0.84 log (lines/1000)	1271.4
E	-0.14 log (lines/100) + 1.19 log (deltas/1000)	980.0
F	1.05 log (deltas/1000)	985.1
G	0.07 log (lines/1000) + 0.95 log (deltas/1000) - 0.44 age	696.3
H	1.02 log (deltas/1000) – 0.44 age	697.4
	Weighted Time Damp	*631.0*

Results Again



Weighted Time Damp Model (Cont.)

- After picking some parameters, they were able to get an error of 631.0
  - ***This was their most successful model***

?

- Exponential (damping) parameter in the time damp model
- Rate at which the contribution of old changes disappears
- Error is minimized with respect to this
  - ***Over different time periods,*** ? ***could differ by a factor of 2***

Any Questions?

Results

TABLE 1
Models to Fit Fault Data

Model	Intcp	Common	Intl	US	Error
(A) Stable	-	-	-	-	757.4
(B) Null model	-	-	-	-	3108.8
(C) Organization only	3.46	0	-0.13	-1.39	2587.7
(D) 0.84 log(lines/1000)	0.92	0	0.17	-0.92	1271.4
(E) −0.14 log(lines/1000) + 1.19 log(deltas/1000)	3.31	0	0.46	-0.70	980.0
(F) 1.05 log(deltas/1000)	2.95	0	0.43	-0.72	985.1
(G) 0.07 log(lines/1000) + 0.95 log(deltas/1000) - 0.44age	2.63	0	0.73	-0.65	696.3
(H) 1.02 log(deltas/1000) - 0.44age	2.87	0	0.74	-0.63	697.4

Weighted Time Damp Model 631.0