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
  - Lines of Code
  - Other Complexity Measures (McCabe, etc.)
    - Highly correlated with lines of code
  - Not very good predictors of faults
- Process Measures
  - Computed from syntactic data
  - Computed from change and defect histories
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

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)

Simulations

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

Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>47.4</td>
</tr>
<tr>
<td>B</td>
<td>108.8</td>
</tr>
<tr>
<td>C</td>
<td>367.7</td>
</tr>
<tr>
<td>D</td>
<td>477.4</td>
</tr>
<tr>
<td>E</td>
<td>900.0</td>
</tr>
<tr>
<td>F</td>
<td>995.1</td>
</tr>
<tr>
<td>G</td>
<td>996.3</td>
</tr>
<tr>
<td>H</td>
<td>107.4</td>
</tr>
</tbody>
</table>

Results Again

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

Observations
Correlation of Complexity Metrics

**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**

<table>
<thead>
<tr>
<th>Model</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>0.377</td>
</tr>
<tr>
<td>Null</td>
<td>0.008</td>
</tr>
<tr>
<td>Organization-Only</td>
<td>0.353</td>
</tr>
<tr>
<td>0.04 log (times/1000)</td>
<td>0.272</td>
</tr>
<tr>
<td>0.05 log (times/1000) + 1.90 log (deltas/1000)</td>
<td>0.360</td>
</tr>
<tr>
<td>0.15 log (deltas/1000)</td>
<td>0.351</td>
</tr>
<tr>
<td>0.07 log (lines/1000) + 0.95 log (deltas/1000) + 0.44 age</td>
<td>0.396</td>
</tr>
<tr>
<td>Weighted Time Damp Model</td>
<td>0.397</td>
</tr>
<tr>
<td>Weighted Time Damp Model (Cont.)</td>
<td>0.321</td>
</tr>
</tbody>
</table>

**Results Again**

<table>
<thead>
<tr>
<th>Model</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.367</td>
</tr>
<tr>
<td>G</td>
<td>0.367</td>
</tr>
<tr>
<td>H</td>
<td>0.367</td>
</tr>
<tr>
<td>WTDM</td>
<td>0.321</td>
</tr>
</tbody>
</table>

**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.
Results

TABLE 1
Models to Fit Fault Data

<table>
<thead>
<tr>
<th>Model</th>
<th>Inter</th>
<th>Omega</th>
<th>fail</th>
<th>pLo</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Null model</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>751.4</td>
</tr>
<tr>
<td>(B) Organization only</td>
<td>2.32</td>
<td>0</td>
<td>-2.13</td>
<td>-0.39</td>
<td>1274.4</td>
</tr>
<tr>
<td>(C) -9.43log(t) + 1.91log(A) + 1.16</td>
<td>3.31</td>
<td>0</td>
<td>0.46</td>
<td>-0.70</td>
<td>108.9</td>
</tr>
<tr>
<td>(D) 1.06log(t) + 1.06log(A)</td>
<td>2.80</td>
<td>0</td>
<td>0.73</td>
<td>-0.72</td>
<td>85.1</td>
</tr>
<tr>
<td>(E) 0.01log(t) + 0.05log(A) + 2.49</td>
<td>2.63</td>
<td>0</td>
<td>0.37</td>
<td>-0.53</td>
<td>93.0</td>
</tr>
<tr>
<td>(F) 0.01log(t) + 0.05log(A) + 2.49</td>
<td>2.87</td>
<td>0</td>
<td>0.74</td>
<td>-0.83</td>
<td>97.4</td>
</tr>
</tbody>
</table>

Weighted Time Damp Model | 631.0 |