Getting More Out of Your Inspection Data: 
Using Capture-Recapture Models for the Reinspection Decision

Julie Barnard
Khaled El Emam
Dave Zubrow

European SEPG Conference
Amsterdam

April 12, 2002
Outline

Background

Capture/Recapture Models and Analytical Approach

Results

Summary and Conclusions
Collaboration Purpose

Investigate the ability of capture-recapture models and analysis to predict remaining defects in software modules after they have undergone an inspection.
Collaboration Tasks

- Provide background information
- Verify and validate data and data definitions
- Conduct experimentation and analyses
- Document results of analyses
- Report results of analyses
- Perform project debriefing
**Design/Code Inspection Process**

- Inspection process generally follows the Fagan style of software inspection
- Inspectors document defects found during pre-meeting individual review in a database
  - Defects found during the inspection meeting are also entered
  - Post-inspection defects are mapped to the inspection that missed them
- An inspection defect is a design or code error that would result in a post-build Discrepancy Report (DR) if left uncorrected and sent to the build
- The project uses inspection defect data and DR data for early detection metrics
  - Early detection metric determines inspection process effectiveness

* Using defined reinspection criteria
Inspection Roles

- Moderator
- Author: Code Developer
- Mandatory Inspectors: Development Peer, Requirements Analyst, Verifier
- Optional Inspectors
- Librarian
A Design/Code Defect is an error that would result in a DR if the error were not fixed pre-build.
Overview of Capture-Recapture Models

Wildlife Ecology Application

• Capture-recapture (CR) models are used in wildlife research to estimate the size of animal populations
  – Animals are trapped, marked, and then released
  – Animals are trapped again (recaptured)
  – Estimates of the animal population are made using information on the number of recaptured animals that are marked

Epidemiological Application

• CR models are used in epidemiology to estimate the size of diseased populations
  – Data from multiple reporting systems are used
Analogy to Software Inspections

• The defects in an inspected document are the animal population

• Each inspector (during the preparation step of the inspection process) represents a trapping occasion

• The data from multiple inspectors are input into a capture-recapture model which is used to estimate the total number of defects in the document

• The inspection team can determine the estimated remaining defects from capture-recapture model output

  – The remaining defects are computed using the total number of estimated defects and the actual number of defects found during the inspection

  – The remaining defect estimate can then be used as criteria for determining necessity of a reinspection
Application of Models to Inspections

- Capture-recapture models applied to software inspections requires:
  - Data to be collected for defects identified by individual inspectors
  - Multiple inspectors to detect at least one defect in common (overlap)

<table>
<thead>
<tr>
<th></th>
<th>Inspector A</th>
<th>Inspector B</th>
<th>...</th>
<th>Last Inspector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect 1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Defect 2</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Defect n</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
</tbody>
</table>

- Selection of the appropriate model for analysis
Classes of Capture-Recapture Models

• There are two general classes of capture-recapture models
  – **Open population models**: population gain (e.g., birth and recruitment) and population loss (e.g., mortality and emigration) occur during the study
  – **Closed population models**: there is no gain nor loss during the study

• We are only interested in *closed* population models

\[
N \text{ (estimated) in work product} = \frac{n(\text{inspector 1}) \times n(\text{inspector 2})}{m(\# \text{ defects found by both inspectors})}
\]

\[
N \text{ (estimated)} - N \text{ (unique discovered)} = \text{Remaining defects (estimated)}
\]
Capture-Recapture Models for Inspections

- **Time Response (t):** On different days animals vary in their catchability
  - Inspectors with different "general abilities" to detect defects

- **Heterogeneity (h):** Different animals vary in their catchability
  - Defects differ in ‘detectability’

<table>
<thead>
<tr>
<th>Model</th>
<th>Inspectors</th>
<th>Defects</th>
<th>Estimators</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>Same defect detection probability</td>
<td>Homogeneous</td>
<td>MLE</td>
</tr>
<tr>
<td>Mt</td>
<td>Different defect detection probability</td>
<td>Homogeneous</td>
<td>MLE, Chao</td>
</tr>
<tr>
<td>Mh</td>
<td>Same defect detection probability</td>
<td>Heterogeneous</td>
<td>JE, Chao</td>
</tr>
<tr>
<td>Mth</td>
<td>Different defect detection probability</td>
<td>Heterogeneous</td>
<td>Chao</td>
</tr>
</tbody>
</table>
Data Source

- Results from 861 design/code inspections were examined
  - Inspected documents ranged in size and complexity
  - Inspections spanned 7 software releases over a period of approximately 7 years
  - Inspection process was essentially the same over the time period
- Downstream defects for the releases were examined
  - The number of downstream defects for older software releases was greater due to the longer field/operational use of the system
    - Several releases had flown assigned Shuttle missions
  - Downstream defects for newer software releases are not yet identified due to the stage of development
Data for Analysis

- 861 inspections, of which 308 had defects

<table>
<thead>
<tr>
<th>Errors Present In Inspection Material</th>
<th>None</th>
<th>One or More</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (553)</td>
<td>Mean (308)</td>
</tr>
<tr>
<td>Inspectors</td>
<td>5.63</td>
<td>6.39</td>
</tr>
<tr>
<td>Inspector Defects</td>
<td>0</td>
<td>1.89</td>
</tr>
<tr>
<td>Meeting Defects</td>
<td>0</td>
<td>2.35</td>
</tr>
<tr>
<td>Overlap</td>
<td>0</td>
<td>1.09</td>
</tr>
<tr>
<td>Total Defects (Inspection + Downstream)</td>
<td>0</td>
<td>4.00</td>
</tr>
<tr>
<td>Yield</td>
<td>N/A</td>
<td>0.57</td>
</tr>
<tr>
<td>Lines of Code (LoC) Changed</td>
<td>135.94</td>
<td>185.50</td>
</tr>
<tr>
<td>Total LoC</td>
<td>1699.17</td>
<td>1655.26</td>
</tr>
<tr>
<td>Preparation Effort</td>
<td>4.13</td>
<td>16.28</td>
</tr>
<tr>
<td>Meeting Effort</td>
<td>3.39</td>
<td>7.07</td>
</tr>
</tbody>
</table>
Data Partitioning

• The following types of inspections were filtered from the analysis:
  – Inspections where no defects are found at all
  – Inspections where only one defect was found
  – Inspections that are already reinspections

• Data from 89 inspections satisfied the criteria for using capture-recapture models
Criterion 1: Ability to Estimate

For the 89 inspections, the number of times that each model was able to produce an estimate is as follows:

<table>
<thead>
<tr>
<th>Model</th>
<th>MtMLE</th>
<th>MhJE</th>
<th>MtChao</th>
<th>MhChao</th>
<th>MthChao</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>49</td>
<td>69</td>
<td>85</td>
<td>55</td>
<td>70</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Model Selection - 2

Criterion 2: Relative Error

- Relative error is computed as

\[ \text{Relative Error} = \frac{\text{Estimated Defects} - \text{Actual Defects}}{\text{Actual Defects}} \]

\[ \text{Estimated Defects} - \text{Actual Defects} \]

\[ \text{Actual Defects} \]

<table>
<thead>
<tr>
<th></th>
<th>M0</th>
<th>MtMLE</th>
<th>MhJE</th>
<th>MtChao</th>
<th>MhChao</th>
<th>MthChao</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median</td>
<td>0</td>
<td>-0.09</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>-0.028</td>
<td>-0.14</td>
<td>0.018</td>
<td>-0.08</td>
<td>0.059</td>
<td>0.1388</td>
</tr>
</tbody>
</table>
Relative Error Distributions
Estimated vs. Actual Defects
Best Fit Model

- MtChao model was highly robust and accurate
  - It was most successful at making estimates
  - It had low relative error
  - Assumes defects have equal probability of detection, but inspectors vary in their abilities to detect defects
Reinspection Decision Results - 1

MtChao model yields 64% correct decision

<table>
<thead>
<tr>
<th>Correct Decision</th>
<th>Pass</th>
<th>Reinspect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Model made Right decision to Pass</td>
<td>Model made Wrong decision to Reinspect</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Model made Wrong decision to Pass</td>
<td>Model made Right decision to Reinspect</td>
</tr>
</tbody>
</table>
Reinspection Decision Results - 2

Logistic Regression model yields **80% correct decision**

<table>
<thead>
<tr>
<th>Correct Decision</th>
<th>Predicted Decision</th>
<th>Pass</th>
<th>Reinspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>29</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Reinspect</td>
<td>9</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

- Model made **Right decision to Pass**
- Model made **Wrong decision to Reinspect**
Reinspection Decision Model

• Logistic Regression model builds on the MtChao decision and incorporates additional inspection attributes
  – Lines of code changed
  – Number of inspectors finding any defects
  – Number of defects found by more than one inspector

• Benefits of using Logistic Regression model
  – Improved accuracy
  – Reduces number of false negative results; i.e., indication to pass when the correct decision is to reinspect
Summary and Conclusions

- Capture-recapture models estimate remaining defects based on inspection detected defects
- CR models enhance the reinspection decision
  - This can supplement existing reinspection decision criteria or
  - This could be used standalone for processes with no existing defined reinspection criteria
- Decisions based on CR models can be augmented through use of additional inspection variables
- CR-based decision models can be institutionalized as part of inspection process
  - Models are a relatively low cost analysis method when used with an existing inspection data infrastructure
References

For general information on capture-recapture models:

– Capture-recapture model software and documentation are available from http://www.cnr.colostate.edu/~gwhite/software.html


Contacts

• Ricardo de la Fuente
  – julie.r.barnard@usahq.unitedspacealliance.com

• Khaled El Emam
  – khaled.el-emam@nrc.ca

• Dave Zubrow
  – dz@sei.cmu.edu