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

USA
United Space Alliance

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.

USA
United Space Alliance

Carnegie Mellon
Software Engineering Institute

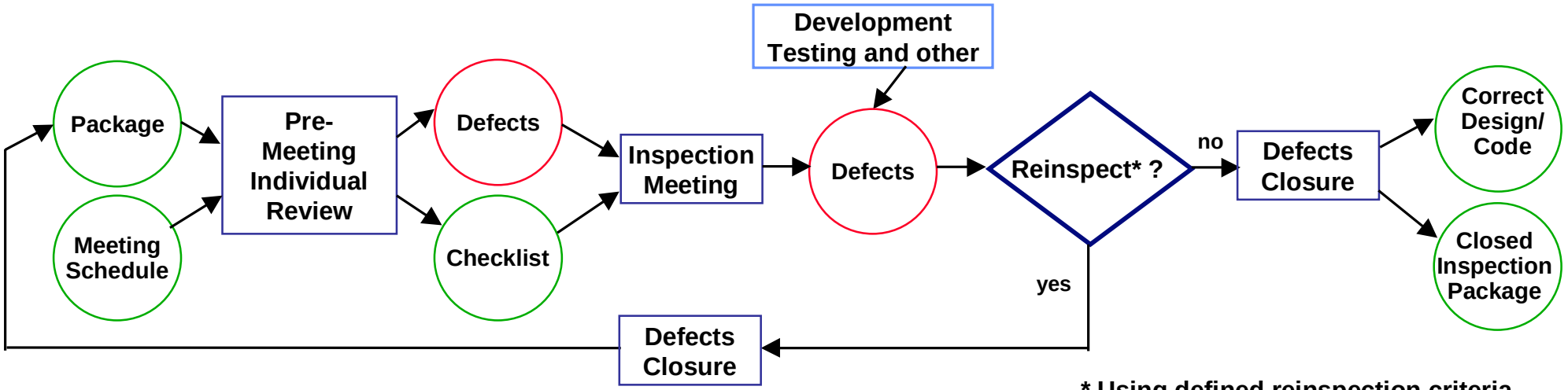
National Research Council Canada

Conseil national de recherches Canada

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

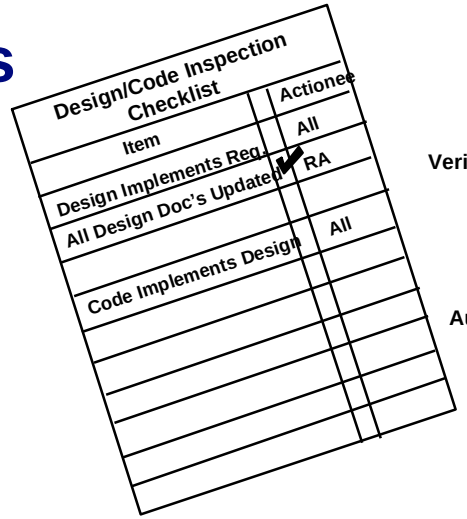
Design/Code Inspection Process

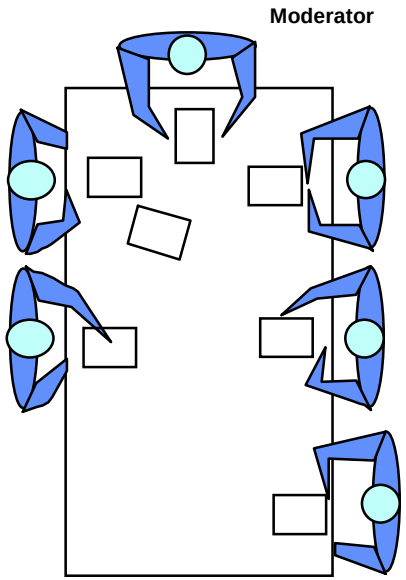


- Inspection process generally follows the Fagan style of software inspection
- Inspectors document defects found during pre-meeting individual review in a database
  - Defects found during the inspection meeting are also entered
  - Post-inspection defects are mapped to the inspection that missed them
- 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

Inspection Roles

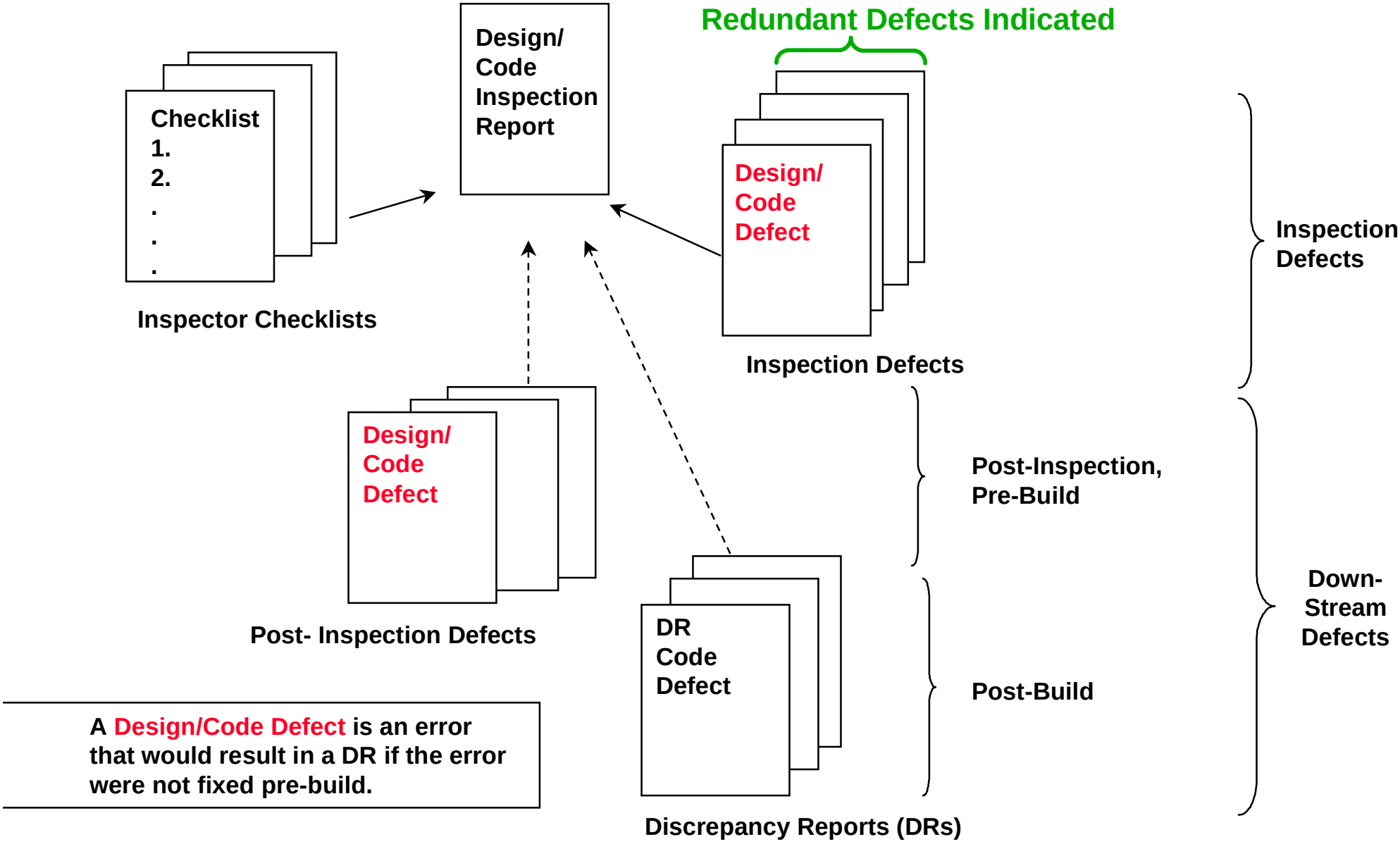
- Moderator
- Author: Code Developer
- Mandatory Inspectors: Development Peer, Requirements Analyst, Verifier
- Optional Inspectors
- Librarian





Librarian

Defect Data Collection



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)

	Inspector A	Inspector B	…	Last Inspector
Defect 1	1	0	…	1
Defect 2	0	0	…	1
…	…	…	…	…
Defect n	1	0	…	0

  - 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 (estimated) in work product =	$\frac{n(\text{inspector 1}) * n(\text{inspector 2})}{m(\#\text{ defects found by both inspectors})}$
N (estimated) – N (unique discovered) =	*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’**

Model	Inspectors	Defects	Estimators
M0	Same defect detection probability	Homogeneous	MLE
Mt	Different defect detection probability	Homogeneous	MLE, Chao
Mh	Same defect detection probability	Heterogeneous	JE, Chao
Mth	Different defect detection probability	Heterogeneous	Chao

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

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

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

Model Selection - 1

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:

M0	MtMLE	MhJE	MtChao	MhChao	MthChao
49	70	69	85	55	70

Model Selection - 2

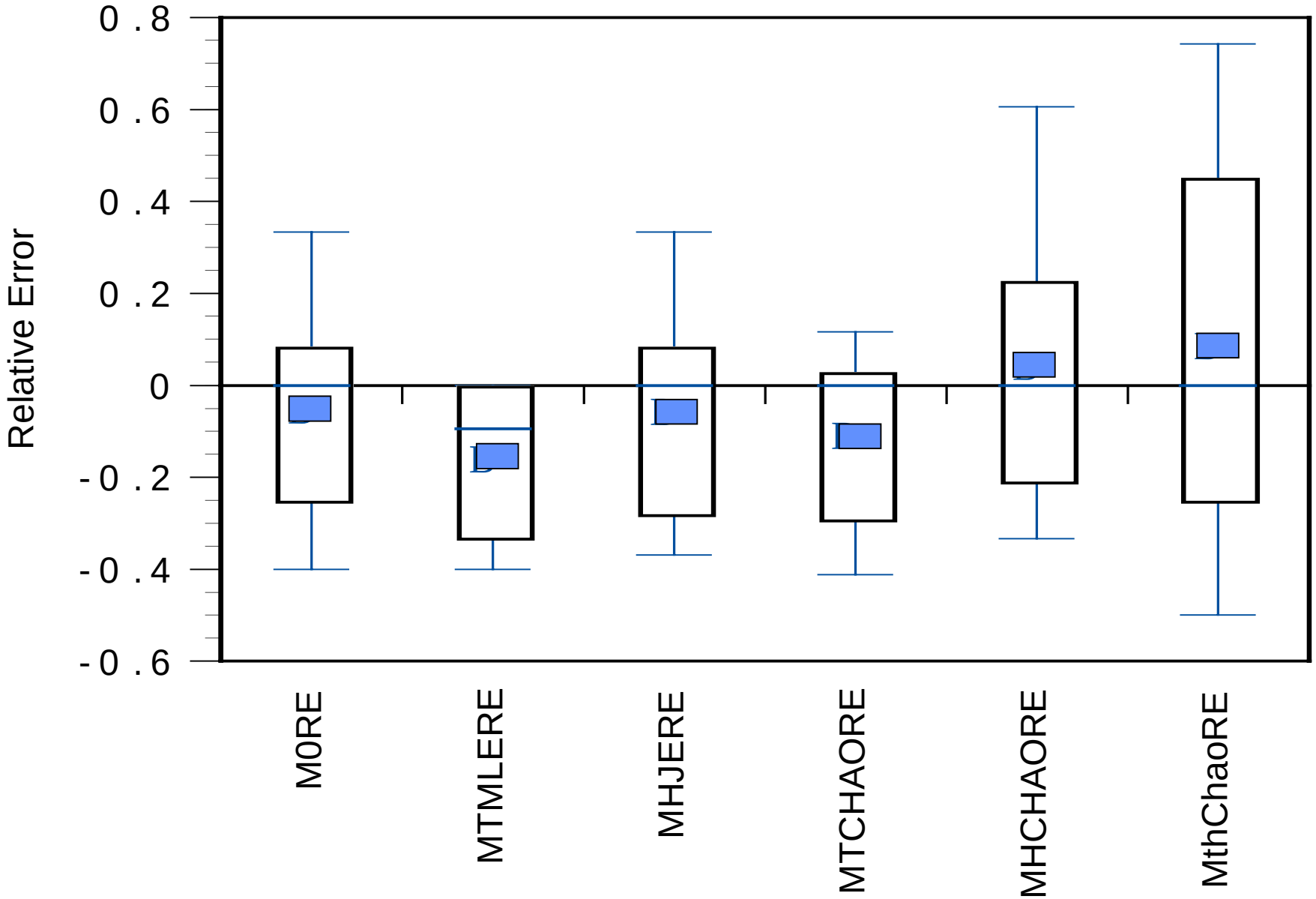
Criterion 2: Relative Error

- Relative error is computed as

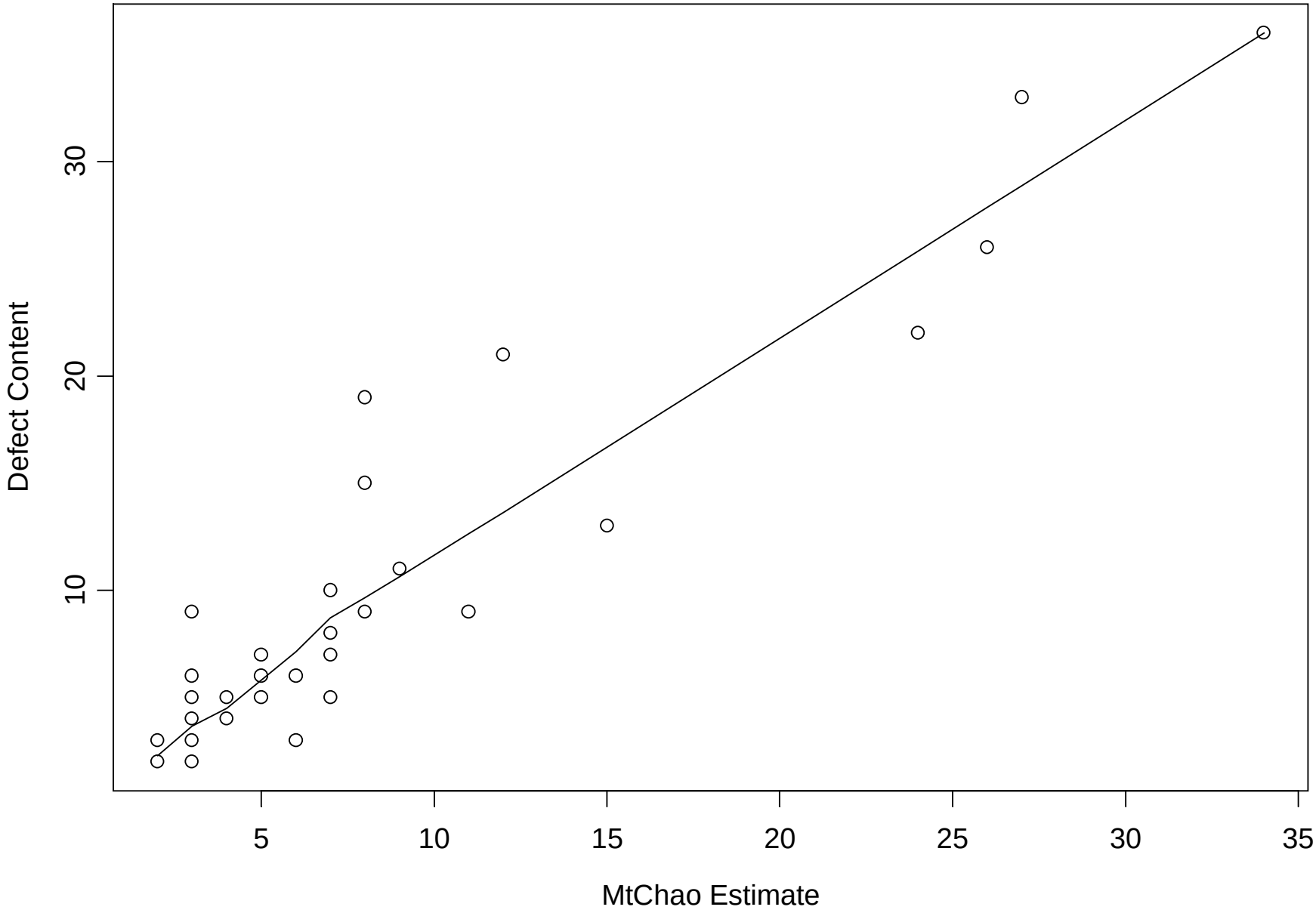
$$\frac{\textit{Estimated Defects} - \textit{Actual Defects}}{\textit{Actual Defects}}$$

	M0	MtMLE	MhJE	MtChao	MhChao	MthChao
Median	0	-0.09	0	0	0	0
Mean	-0.028	-0.14	0.018	-0.08	0.059	0.1388

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

		Predicted Decision: Pass	Predicted Decision: Reinspect
Correct Decision: Pass		**33** Model made Right decision to Pass	**3** Model made Wrong decision to Reinspect
Correct Decision: Reinspect		**25** Model made Wrong decision to Pass	**16** Model made Right decision to Reinspect

Reinspection Decision Results - 2

Logistic Regression model yields 80% correct decision

Correct Decision		Predicted Decision: Pass	Predicted Decision: Reinspect
	Pass	**29** Model made Right decision to Pass	**6** Model made Wrong decision to Reinspect
	Reinspect	**9** Model made Wrong decision to Reinspect	**32** Model made Right 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
- L. Briand, K. El Emam, B. Freimut, and O. Laitenberger: "A comprehensive evaluation of capture-recapture models for estimating software defect content". *IEEE Transactions on Software Engineering*, 26(6):518-540, June 2000.
- K. El Emam and O. Laitenberger: "Evaluating capture-recapture models with two inspectors". *IEEE Transactions on Software Engineering*, 27(9):851-864, 2001.
- Humphrey, W. Introduction to the Team Software Process. Reading, MA: Addison Wesley, 2000.
- Petersson, H. and Wohlin, C. "An empirical study of experience-based software defect content estimation methods" Lund University.

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