Experiences with Capture-Recapture

Darryl L. Davis
Davis Systems

Noopur Davis
Software Engineering Institute

SEPG 2005
Seattle, Washington

Agenda

- Introduction
- Experiences
- Conclusion

Introduction

Finding & Removing Defects

- We review, inspect, and test our engineering work products to find and remove defects.
  - We would like to find and remove them all.
  - We can easily keep count of how many we've found.
  - But how can we determine if there are any left, and how many?

The Capture-Recapture Method

- Watts Humphrey had the idea of a simple way to project remaining defects based on the data from two or more independent reviewers (using peer reviews/inspections or tests to find defects).
- He later found that such a method is used in estimating animal populations, and is called "capture-recapture."

How Many Fish?

- You catch 30 fish from a pond, tag them, and release them.
- A few days later, you catch 25 fish from the pond. 5 of them have tags.
- About how many fish are in the pond?

Calculating the Total

$$\frac{30 \text{ tagged fish (in pond)}}{x \text{ total fish (in pond)}} \cong \frac{5 \text{ tagged fish (in sample)}}{25 \text{ total fish (in sample)}}$$

$$x \text{ total fish (in pond)} \cong \frac{30 \text{ tagged fish (in pond)} \times 25 \text{ total fish (in sample)}}{5 \text{ tagged fish (in sample)}}$$

$$x \text{ total fish (in pond)} \cong 150$$

Application to a Product

- We want to determine the total number of defects in a product.
- The product is the “pond.”
- Defects are the “fish.”

Estimating Remaining Defects

- Two engineers conduct independent appraisals of a product and record all the defects they find.
- Those found by the first engineer are the “tagged” defects.
- The second engineers typically finds some of the same “tagged” defects, as well as others.
- Just as with the fish, we can estimate the remaining defects (the ones not found by either engineer).

The Capture-recapture Method*

1. Count the number of defects recorded by the first engineer (A)
2. Count the number of defects recorded by the second engineer (B)
3. Count the number of defects found by both engineers (C)
4. Estimate the total defects as (A*B)/C
5. Calculate the defects found as (A+B)-C
6. Estimate the remaining defects as (A*B/C) - (A+B-C)

* [Humphrey 2000]

With Multiple Inspectors

For inspections with more than two engineers:

- Count the defects recorded by the engineer who found the most unique defects (A).
- Combine the defect data from the rest of the inspectors (B). When multiple engineers find the same defect, count it just once.
- The rest of the calculations remain as for the two engineer case.

Experiences

A Real World Example

Product Size Reviewed (Requirements Statements)		357
Major Defects Found		42
Defect Projection (Major Defects Only)		
	Reviewer(s) Finding Most Unique Defects	1 & 2
	Unique Defects Found by Reviewers 1 & 2	11
	Total Defects Found by Reviewers 1 & 2	31
	Total Defects Found by the Other Reviewers (3-7)	31
	Defects Found by Both Groups	20
	Total Defects Projected	48
	Total Defects Found	42
	Remaining Defects Projected	6

A Real World Example (continued)

Defect Projection (Counting only Major Defects)			
	Total Defects Projected	48	
	Total Defects Found	42	
	Remaining Defects Projected	6	
Estimated Yield (Counting only Major Defects)			
	Defects found by Reviewer 3	23	
	Defects found by Reviewer 4	12	
	Defects found by Reviewer 5	1	
	Defects found by Reviewer 6	13	
	Defects found by Reviewers 1 & 2	31	
	Defects found by Reviewer 7	3	
	Yield by Reviewer 3	48	Percent
	Yield by Reviewer 4	25	Percent
	Yield by Reviewer 5	2	Percent
	Yield by Reviewer 6	27	Percent
	Yield by Reviewers 1 & 2	65	Percent
	Yield by Reviewer 7	6	Percent
	Total Inspection Yield	87	Percent

A Real World Example (continued)

- This team went on to find most, but not all, of the projected 6 major defects prior to acceptance test.
- Only one defect was found in acceptance test: a requirements defect, as projected.

An Experiment

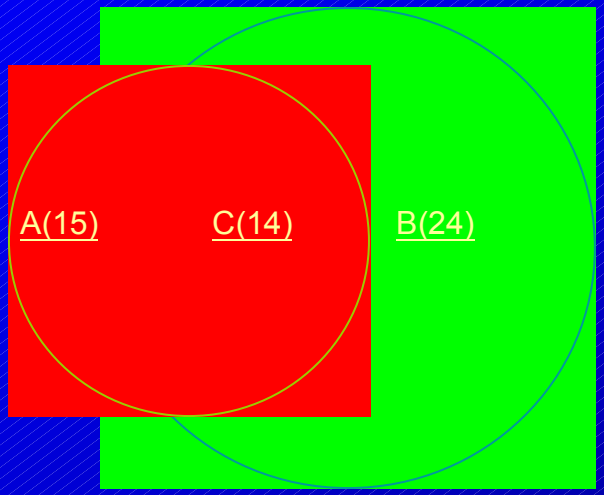
- We took an existing application
  - 708 LOC
  - Developed using personal reviews and inspections
  - In use for over two years
- We seeded this with 26 defects
- Then, we conducted an inspection with a team of eight developers

Inspection Summary Data

	Defects		
Engineer Name	Major	Minor	Estimated Yield
Engineer 1	13	3	50%
Engineer 2	15	2	58%
Engineer 3	22	2	85%
Engineer 4	15	1	58%
Engineer 5	13	2	50%
Engineer 6	15	5	58%
Engineer 7	13	2	50%
Engineer 8	16	1	62%
Total			96%

	Total Major			15	24	14	13	15	22	15	13	15	13	16
	Unique Defects						1	0	0	2	0	0	0	0
No	Defect Description	Major	Minor	A	B	C	e1	e2	e3	e4	e5	e6	e7	e8
1	**File 1**													
2	19 - session variables should be in application on-start	x		x	x	x	x	x	x	x	x		x	
3	17, 20 - https instead of http	x			x				x		x	x	x	x
4	Option explicit not used	x			x		x							
5	21 - session variables not initialized	x		x	x	x		x	x	x	x	x	x	x
6	**File 2**													
7	1 - Explicit use of scrip language	x			x		x	x	x		x	x		x
8	12 - Config.asp is not run again	x		x	x	x				x				
9	**File 3**													
10	14 - session hijaking issue	x		x	x	x		x	x	x		x	x	x
11	14 - forward slash/back slash	x			x		x	x	x		x	x	x	
12	**File 4**													
13	3 - GET method is used	x			x			x	x				x	x
14	8, 12 - size does not match variable	x		x	x	x	x		x	x	x	x	x	x
15	**File 5**													
16	7 - No user validation	x		x	x	x	x	x	x	x	x	x	x	x
17	15 - s_userID is set before validation	x		x	x	x		x	x	x		x	x	x
18	21- Error handling is an issue	x		x	x	x	x		x	x	x	x	x	x
19	23 - SQL injection	x		x	x	x	x	x	x	x				
20	31 - possible to store passwords in plain text	x		x	x	x		x	x	x		x		x
21	49 - Possible overflow of long integre	x			x			x	x					
22	55 - Atomic transaction needed	x			x		x	x	x					x
23	60 - the password field is overloaded	x			x		x	x	x					
24	60 - default passwords are bad	x		x	x	x		x	x	x	x	x	x	x
25	66 - logedIn variable is overloaded (changepass)	x		x						x				
26	77 - loop is off by one	x			x		x		x		x	x		x
27	107 - Write is off	x			x		x		x		x	x	x	x
28	110 - Too much error information	x		x	x	x	x	x	x	x	x	x	x	x
29	68-98 - can't get admin priv if not first login	x		x	x	x			x	x	x	x		
30	**File 6**													
31	12 - Cross Site Scripting	x		x	x	x			x	x				x

Capture-Recapture Data



Product Size	708	
Size Measure	LOC	
Total defects found for (A)	15	
Total defects found for all others (B)	24	
Number of Defects Found in common (C)	14	
Estimated Total Major Defects (AB/C)	26	
Number of Major Defects Found (A+B-C)	25	
Number of Major Defects Remaining	1	
Number of Minor Defects Found	5	
Meeting Time (minutes)	60	
Total Meeting Effort (hours)	8.00	
Total Inspection Hours	14.00	
Overall Review Rate	32.2	
Defects/Hour	1.1	
Estimated Defect Density	36.7	per KLOC

Results of the Experiment

- The team found 25 of the 26 defects we had seeded.
- Capture re-capture predicted 1 remaining defect.

Conclusion

Cautions

- As with all statistical methods, or whenever data is used to make decisions, use caution.
- Use the results only as guidance.
- Capture re-capture works best when
  - The number of defects found is not too small.
  - The defect detection is good at finding all of the defect types that are being projected.
  - All reviewers achieve fairly high yields.

Summary

- Capture-recapture is a simple and effective way of projecting the remaining defects in a product and evaluating the defect-finding activities.
- It provides information needed for product quality and process control decisions.
- Applied appropriately, it works!

References

- [Humphrey 2000] Watts Humphrey, *Introduction to Team Software Process*, Addison Wesley, 2000.

Voice: +1 (412) 683-1921
Fax: +1 (877) 817-7639
E-mail: DDavis@DavisSys.com
Web: http://www.DavisSys.com

- Trademarks
  - Carnegie Mellon, Capability Maturity Model, CMM, and CMMI are registered in the US Patent and Trademark Office by Carnegie Mellon University.
  - SEI, Team Software Process, TSP, Personal Software Process, PSP, and SCAMPI are service marks of Carnegie Mellon University.