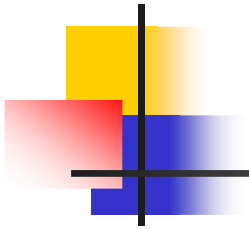
ÉCOLE POLYTECHNIQUE MONTRÉAL

SOCCER LAB
SOftware Cost-effective Change and Evolution Research Lab

Threats on Building Models from CVS and Bugzilla Repositories: The Mozilla Case Study

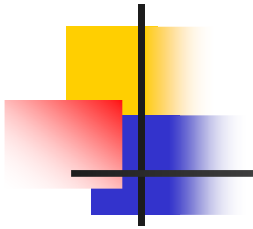
K. Ayari, P. Meshkinfam, G. Antoniol and M. Di Penta

Outline



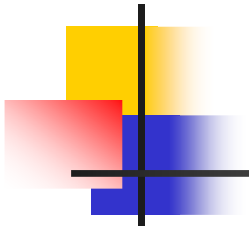
- ☑ *Versioning systems and problem reporting systems*
- ☑ *Why software change classification is a problem ?*
- ☑ *Our approach*
- ☑ *Results*
- ☑ *conclusion*

CVS and Bug Tracking System



- CVS/SVN and bug tracking systems are the backbone of **open source** and widely adopted by the **industry**
- Bug tracking systems do not only contain "defect" related information (there is much more)
- Bug tracking document changes
  - perfective
  - preventive
  - "real" bug fixing
  - communication between developers
  - legal and copyright matters
  - Etc.

CVS/SVN



- Documents the history of changes
- Effective to document code changes but…

  poor in enforcing **traceability** with the **reason of the changes**
- Traceability links are mostly documented on a voluntary basis!
- Log messages in commit information often refer the issue id (bug id) root cause of the changes

Why Bug Tracking is Important?

- Bug tracking systems contain a wealth of information related to software quality but …entry classification is often
  - unclear
  - imprecise
  - and even wrong!

- This information is vital to build models of software quality

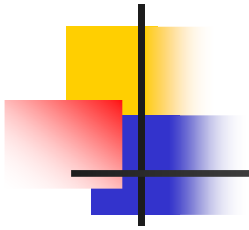
But...

Are we really modeling defects ?

Traceability Definitions - IEEE SE Glossary

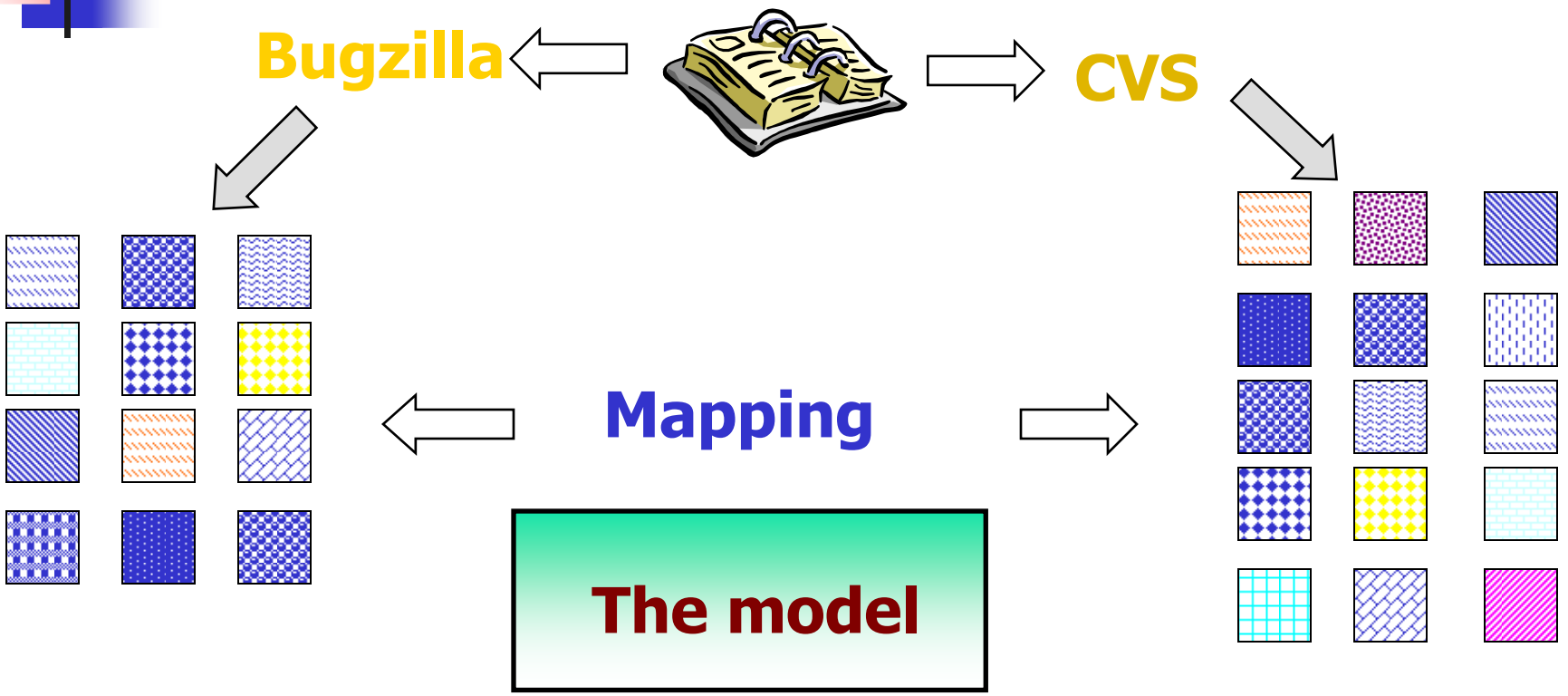
- The **degree** to which a **relationship** can be established **between two or more products** of the development process, especially products having a **predecessor-successor** or **master-subordinate** relationship to one another;
  - the degree to which the **requirements** and **design** of a given software component **match**;
- The degree to which each element in a software development product establishes its **reason for existing**;
  - the degree to which **each element in a graphical environment** references the **requirement that it satisfies**.

Where is the Problem?

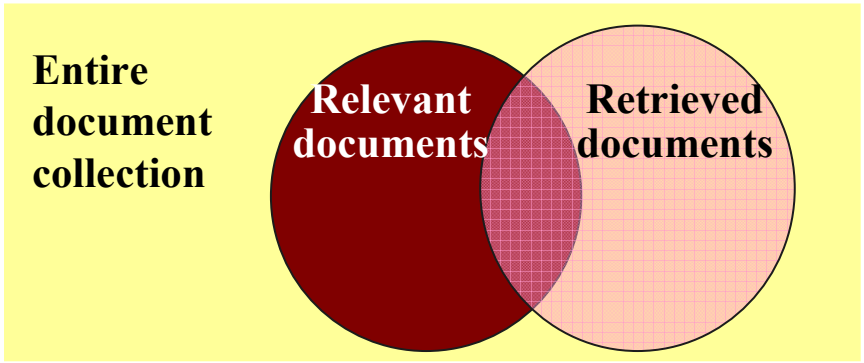


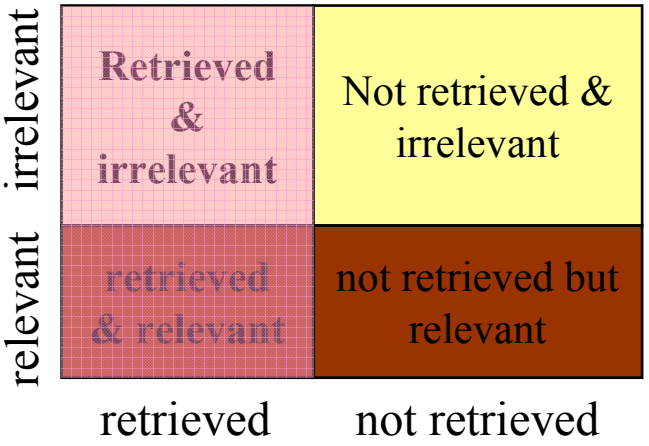
- We need to recover traceability links between
  - source changes documented in CVS/SVN
  - bugs as logical entities as documented in bug tracking systems
- Traceability links are not there
- Traceability recovery approaches are available but :
  - with unknown accuracy
  - are often too rigid
  - they assume that bug tracking systems only contain bugs

A Traceability Model



Accuracy - Precision and Recall

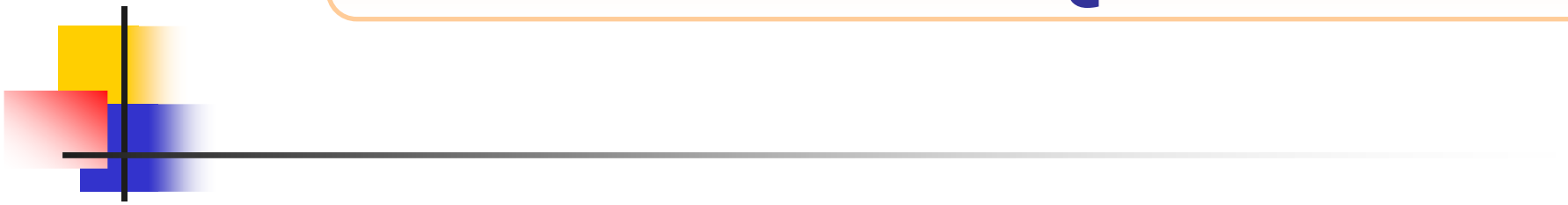




$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

Two Research Questions

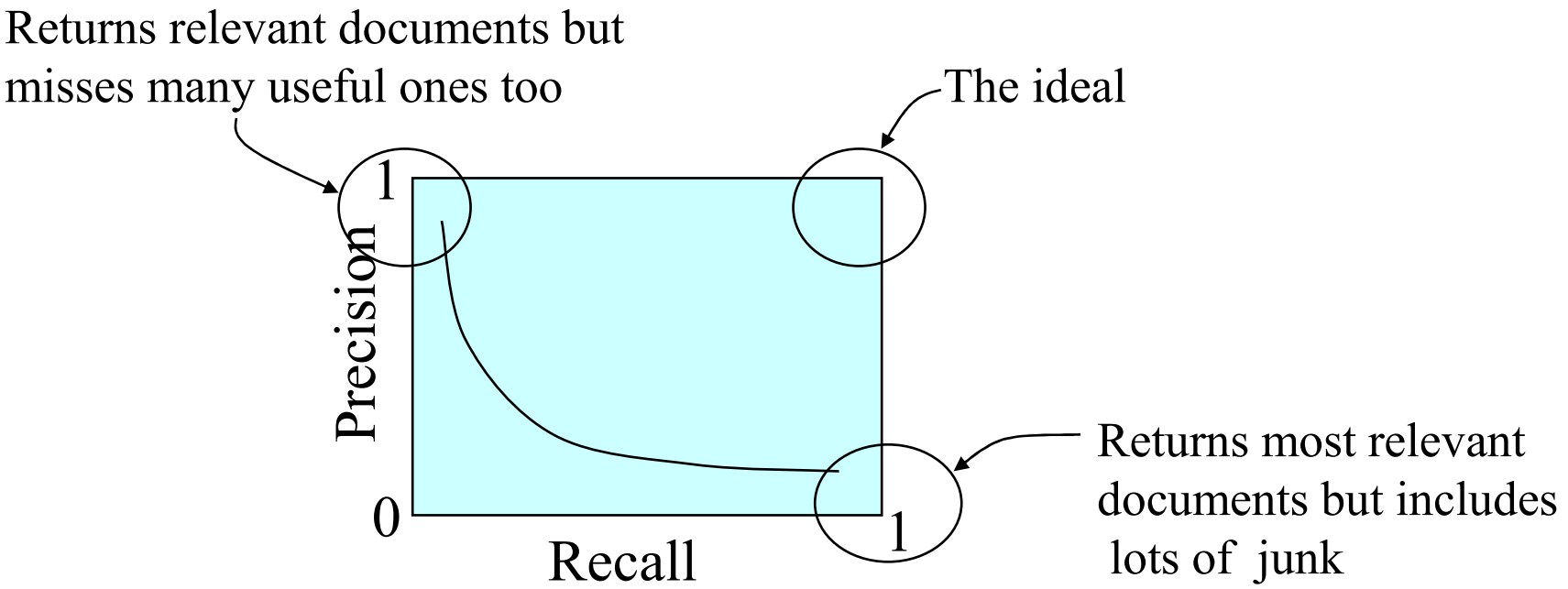


- Accuracy in traceability link recovery and bug tracking system content may **challenge** the applicability of quality models built from CVS/SVN and bug tracking systems

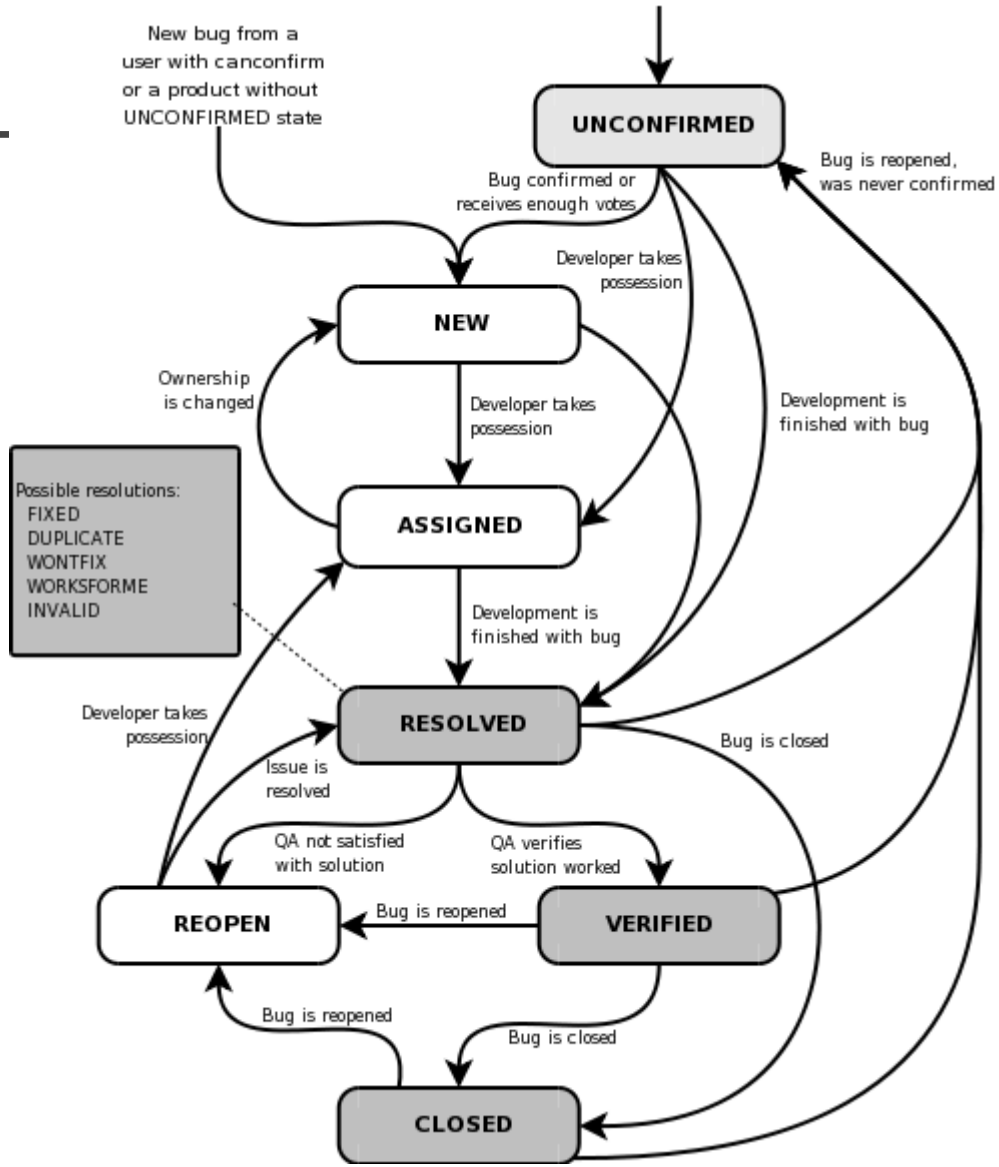
- RQ1: What is the accuracy (i.e. **precision** and **recall**) achieved when recovering traceability links with available methods?

- RQ2: To what extent **Bugzilla's entries *correspond*** to **corrective maintenance** interventions?

Trade-off between Recall and Precision

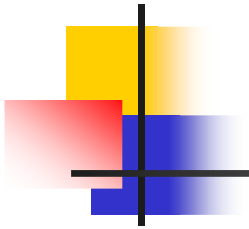


Bugzilla Bug Life-cycle



Great, but is it consistently applied?

Traceability Recovery



- Based on regular expression
  - matching between CVS/SVN log messages and BUG IDs
- a simple regular expression :
  - [\s#=][0-9]{4,6}[!:;.\?\.\s\]\)\-$]
- a complex regular expression :
  - ((b)[ug]{0,2}\s*[id]{0,3}|id|fix|pr|#) [\s#=]*\[([0-9]{4,6}\]?

Plus:

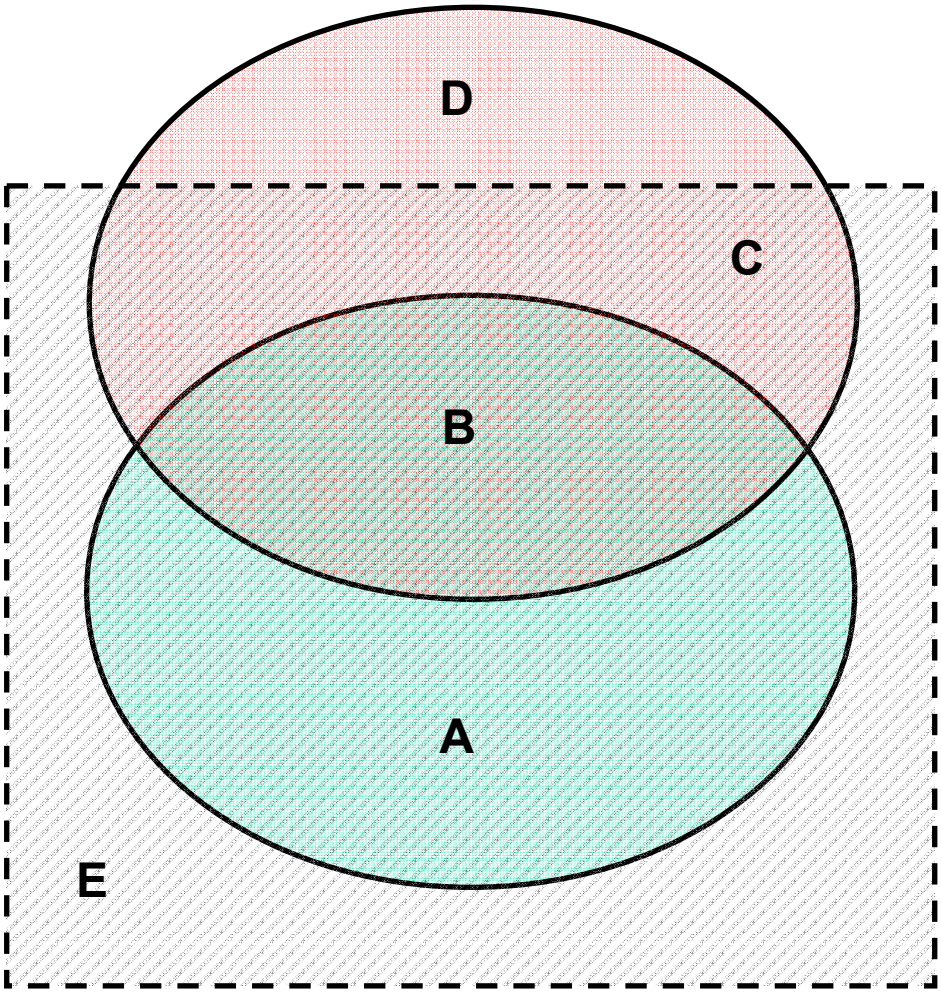
- fix(e[ds])?|bugs?|problems?|defects?|patch"

Possible Matches

Matched from CVS Messages

Matched from Patches

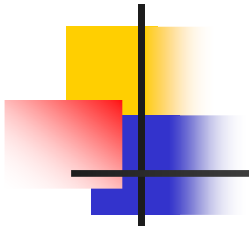
Issues on Bugzilla



Mozilla Bugzilla Facts

- Bugs are almost never closed
  (few hundreds out of about 98500 entries)
- There are a few duplicated, not a bug, won't fix entries
- Most relevant to us
  - VERIFIED and FIXED as well as
  - RESOLVED and FIXED

Status VERIFIED versus RESOLVED



- Resolution FIXED versus INVALID, WONTFIX, WORKSFORME, DUPLICATED and MOVED
- VERIFIED: bugs that have an approved resolution
- RESOLVED: a bug resolution is proposed

Mozilla Bugzilla Recall

$$recall = \frac{Number\ of\ \ relevant\ documents\ retrieved}{Total\ \ number\ of\ \ relevant\ documents}$$

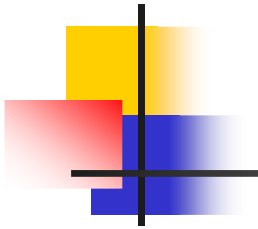
	Retrived links	In Bugzilla	Recall
RESOLVED	19,640	47,163	42 %
VERIFIED	15,482	45,695	34 %
Overall	35,122	92,858	38 %

Matching per Type/Area

	RESOLVED	VERIFIED	OVERALL
A	9,499	7,992	17,291
B	14,730	10,997	25,727
C	4,910	4,485	9,397
D	3006		
E	18,024	22,212	40,236

Resolved-Fixed Vs Verified-Fixed

A Manual Classification



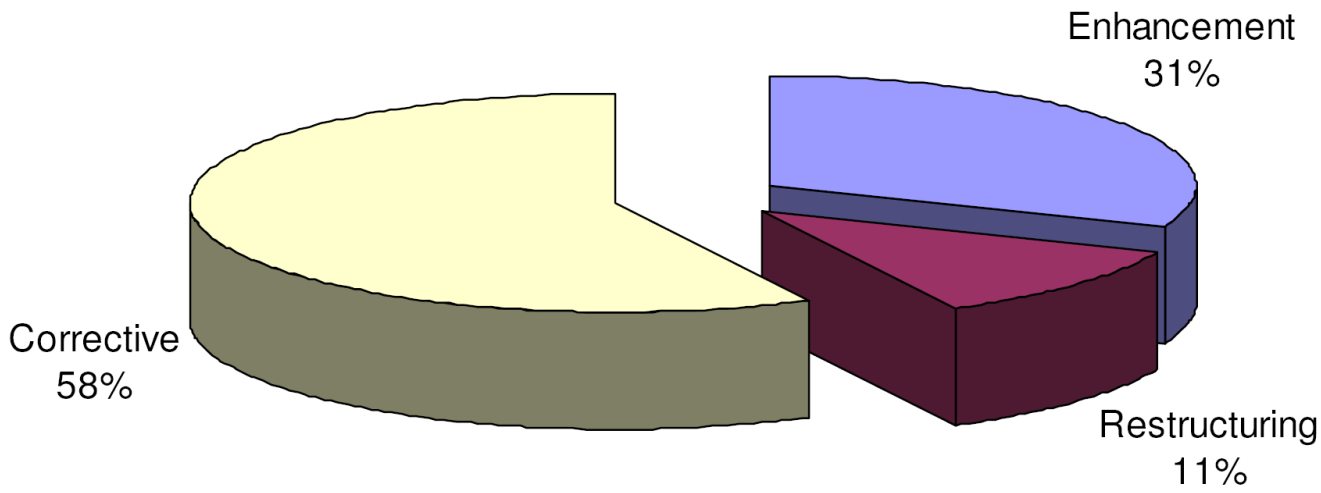
In order to isolate the "Real Bugs" :

- There are about 35000 retrieved links
- Exhaustive manual verification unfeasible
- Select a sample size to ensure a confidence level of 95% and a confidence interval of 10%.
  - 600 links
  - 4 people classified each 150 links
  - Consensus building on each of 600 likns

Manual Classification Taxonomy

- ***Corrective***, i.e., real bugs, issues related to corrective maintenance
- ***Enhancements***, i.e., issues related to the addition of new features, improvement of existing ones, performance increase, GUI layout enhancement, etc.;
- ***Restructuring***, related to code refactoring without altering its behavior, addition of licenses, comments, etc.

Defect Manual Classification



Conclusion

- Out of about 90,000 issues, RESOLVED or VERIFIED with resolution FIXED **only 38%** are traced into CVS modifications
- Only a subset of issues (about a half) posted on Bugzilla are real defect, i.e., corrective maintenance

Models built with current traceability recovery practices are likely NOT to predict quality as defect proneness

Thank you for your attention

questions?