Developer-related factors that correlate with bug density:
An empirical study of the Linux kernel

Philip J. Guo
Computer Systems Laboratory
Advisor: Prof. Dawson Engler

Computer Forum Poster Session
March 18, 2008
Motivation

There is an abundance of common wisdom and expert opinions on what practices lead to higher-quality software but a lack of quantitative scientific validation of these claims.
Objective

To quantify relationships between developer-related properties and software quality by performing an empirical study of the Linux kernel.
Methodology

- Data mining and statistical analysis on Linux source code history (2002-2007)
- Use bug density (# bugs/KLOC) as surrogate for software quality
- Correlate developer-related properties with bug density
Bug populations

- **22,283** out of **123,512** patches are *bugfix patches*
  - Suffer from selection bias and misclassification

- **2,125** bug reports from *Coverity Scan*
  (automated tool finds bugs in open source projects)
  - Suffer from false positives (36%) and limited scope of automated tools
Control for development activity

Figure 1: Num. non-bugfix patches vs. bug density

Figure 2: Num. developers vs. bug density

More eyeballs might mean more bugs found and fixed (or maybe just more bugs)

Figure 3: Num. developers vs. bug density, controlling for num. patches (values in gray fail Spearman test at $p < 0.01$)
Developer populations

- **4,131** developers active in 2002-2007
- **Top 10%** most active (413) wrote **86%** of total patches
- **Top 1%** most active (41) wrote **45%**
- **Professionals** (1,840) wrote **65%**
  - 62% of Top 10% and 66% of Top 1% are pros
Experienced developers write less buggy code and find more bugs

Figure 4: Percentage of patches by top 10% devs. vs. bug density, controlling for num. patches

Figure 5: Percentage of patches by top 1% devs. vs. bug density, controlling for num. patches
Professional developers write less buggy code and find more bugs

Figure 6: Percentage of patches by professional devs., controlling for num. patches
More focused and active development leads to more bugs

Median number of files affected per patch

% patches that only affect this single file
Greater variation in contributions is better for detecting bugs

Percentage of patches by lead developer

Coefficient of variation in num. patches per developer
The mixed effects of temporal variation in patches

% longest streak of patches by one developer

Number of transitions per developer