Architecture-based Speculative Analysis to Predict Bugs

Duc Le – University of Southern California

Explicitly manifest as bugs

Problems

Implicitly result in “smells”

Bug history

Underlying architecture

Explicitly manifest as bugs

Problems

Implicitly result in “smells”
How to use architecture?

Architectural recovery

Find “smells”

- Duplicate Functionality
- Logical Coupling

- Scattered Parasitic Functionality
- Concern Overload

- Dependency Cycle
- Link Overload

- Ambiguous Interface
- Unused Interface
- Unused Brick
- Sloppy Delegation
- Brick Functionality Overload
- Lego Syndrome

Multiple viewpoints

Architectural recovery

Find “smells”
Are “smells” responsible for breaking your system?

Smelly issues => buggy
Smelly classes => issue- & change-prone

Correlate smells and issues

Track
• Architectural smells

Correlate
• Smells vs. issues

Build
• Prediction models

Thesis overview