Software Quality: Moving Research To Practice

Ahmed E. Hassan
NSERC / RIM Research Chair
in Software Engineering of
Ultra Large Scale Systems
Queen’s University, Canada

http://sail.cs.queensu.ca
Intelligence throughout the lifetime of a software system
From inception to production

http://sail.cs.queensu.ca
Can SAIL help?!
We got a paper for that!
Uneven participation in discussions is worrisome

Participation of experienced developers is not a good sign
Missing to update code comments: Not as big of a problem

Inconsistent comment updates is the problem!
A Complex Code Development Process Impacts Code Quality

Low Entropy

High Entropy

[ ICSE 2009]
Building on Fresh Code is Dangerous!
Uneven participation in discussions is worrisome
Participation of expert developers is not a good sign

Missing to update comments:
Not as big of a problem

Inconsistent comment updates is the problem!

A Complex Code Development Process Impacts Code Quality

Building on Fresh Code is Dangerous
Interesting

Makes sense!
Data Noise

Bias exists in the best industrial data sets

[WCRI 2010]

Inconsistent Results

Analysis level matters (files vs. subsystems)

[ICSM 2010*2]

Why no adoption?
How can we move our research to practice?!
Timely
Explainable
Assignable
Two examples of industrial adoption

- Risk Analysis of Code Changes
- Automated Analysis of Load Tests
How risky is a code change?

Changes

Risk Classifications

 Characteristics of Risky Changes

New Code Change

Accept it!  Closer Review  More Testing
As accurate as developers risk.

BONUS:
Provides rationale for risk
(e.g. complex change, too spread out)

Automated Risk

One Year of Changes

450 developers
60 teams

0.84

Developer Risk

[ FSE 2012]
Timely
Explainable
Assignable
Two examples of industrial adoption

Risk Analysis of Code Changes

Automated Analysis of Load Tests
Most field problems for large systems are rarely functional instead they are load-related
Mimics multiple users repeatedly performing tasks for hours or even days

Produces GB/TB of data that must be analyzed
Automated Verification of Load Tests

Fact:
Load testing repeatedly executes the same scenarios

Intuition:
Dominant behaviour indicates normal operation
Minority behaviour is likely problematic
Automated Verification

• (E2, E3) are always together:
  – (acquire_lock, release_lock)
  – (open_inbox, close_inbox)

• If we see (E2, E6) this might be a problem
Report for Dell DVD Store

99.99% reduction in viewed log lines with a precision of 56-100%
Timely
Explainable
Assignable
Timely
Explainable
Assignable

Two examples of industrial adoption

Risk Analysis of Code Changes

Automated Analysis of Load Tests

Software Analysis & Intelligence Lab

http://sail.cs.queensu.ca

http://MSRConf.org
http://promisedata.org