

# Verification and Validation for Concurrent Programs

A State-of-Practice Questionnaire

Margaret A. Wojcicki  
School of Information Technology and Electrical  
Engineering, University of Queensland  
e-mail: wojcicki@itee.uq.edu.au

## Overview

- Background
- Related Work
- Survey Methodology
- Survey Results
- Survey Analysis
- Conclusions and Future Work

## Background

- Verification: does a computer program conform to its specification?
- Validation: does a computer program meet the requirements of the client?
- Verification and validation (V&V): dynamic or static
- Cost-effectiveness factors: types of defects, cost of defect detection and false-positives (mostly static)

## Background

- Empirical studies provide information on V&V technologies
- Range from controlled experiments to surveys
- Survey provides practitioner input
- 'real world' answers to research questions

## Related Work

Survey	Methodology	Principal Results
1. V&V Australia ('04)	Online, telephone and interviews	Low automation
2. V&V Sweden, US and Multinational ('03)	Online	Low automation Low unit testing
3. V&V Canada ('04)	N/A	Low automation Low unit testing
4. V&V Sweden ('02)	Workshop and interview sessions	Larger companies more systematic and use commercial tools
5. Software development NZ ('00)	Telephone and in person	Larger companies more systematic
6. Software Testing Institute ('02)	Online	Low automation Online sources of V&V technology information
7. Quality Assurance Institute ('02)	At Intl. Conference on Software Testing	Code reviews are most common static test

## Survey Methodology

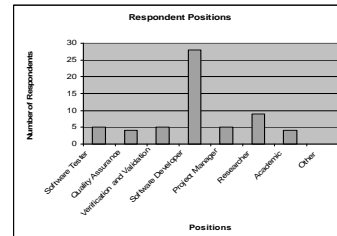
- Objectives:
  1. Examine current V&V technology use and context of use
  2. Examine decision process for V&V selection
- Survey sections:
  1. Context of V&V Technology
  2. Source of information
  3. Case Study Aspects
  4. Cost-effectiveness factors

## Survey Methodology

- Method:
  - self-recruited +cross-sectional = non-systematic sampling
- Sample selection (convenience):
  1. Concurrency-interest mailing list
  2. IBM developerWorks Multithreaded Java Programming forum
  3. Snowballing
  4. 5% response rate adequate (Lethbridge et al)

## Survey Results - Demographics

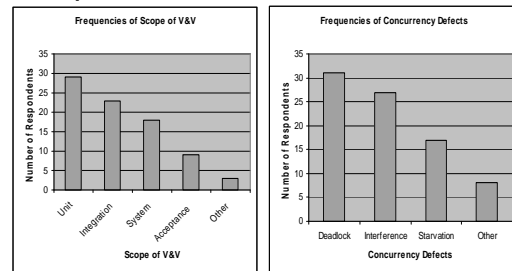
- 35 responses
- Majority have at least 3 years of V&V experience



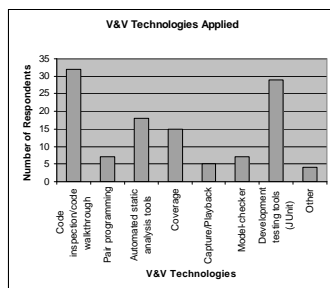
## Survey Results – Programming Languages

- Predominantly Java, but also C++ (smaller numbers using Ada and C)
- average of 7.3 years experience with Java
- currently using version 1.5 or more recent

## Survey Results – Scope and Defects



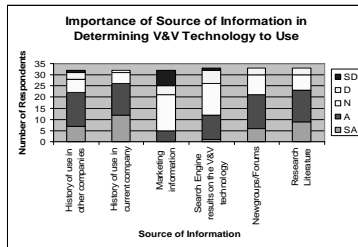
## Survey Results – V&V



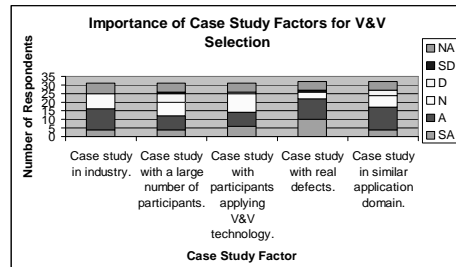
## Survey Results – V&V

- Code inspection/code walkthrough predominantly done ad-hoc and individually
- 60% use open source technology: mostly JUnit, FindBugs

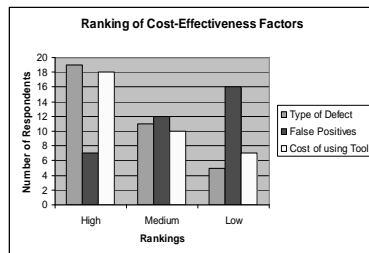
## Survey Results – Decision Process



## Survey Results – Case Studies



## Survey Results – Cost-effectiveness



## Survey Analysis – Needs

- V&V technology:
  - Java 1.5, C++
  - Code inspection/code walkthrough (automation)
  - Systematic inspections (groups)
  - V&V technologies (coverage tools and model-checkers)

## Survey Analysis - Needs

- Case studies:
  1. Real defects (bug repositories, benchmarks)
  2. Measurable results
- Decision Process:
  - History in Industry ↔ Research Literature

## Survey Analysis

- Cost-effectiveness:
  - Willing to deal with false-positives (quality reports)
  - Also: code conversion, implementation/maintenance of tool, purchase cost relative to defects caught
- Previous surveys:
  - Low Automation
  - Unit testing versus acceptance (surveys 2 & 3)

## Survey Analysis

- Threats to Validity:
  - Low internal, but high external validity
  - Generalise to software developers, but no control on background/knowledge
  - Non-response bias (difficult to calculate)
  - Convenience sampling
- Other issues:
  - Definition of V&V (correctness)
  - Sources of Information question biased to companies
  - Ranking question (defaults to 1)

## Conclusions

- V&V mostly manual, aided with development testing tool automation (JUnit)
- Importance of real defects in case studies
- Many aspects of cost-effectiveness
- Qualitative information improves understanding of context

## Future Work

- Examine locking mechanism used by practitioners
- Focus on performance issues rather than correctness and debugging tools
- In depth examination of C and C++ (memory management)
- Modification of TestCon and schema

## Questions and Answers

