

Testing: A Roadmap



Mary Jean Harrold
Georgia Tech

Supported by Microsoft, Inc., Boeing Commercial Airplane Group, Lucent Technologies, Inc., and National Science Foundation



ICSE 2000 Future of Software Engineering

1

Testing



Input	Exp Output	Act Output
1 2	1.5	1.5

- Determine coverage
- Measure performance
- Determine pass/fail
- Consider actual output as behavioral specification

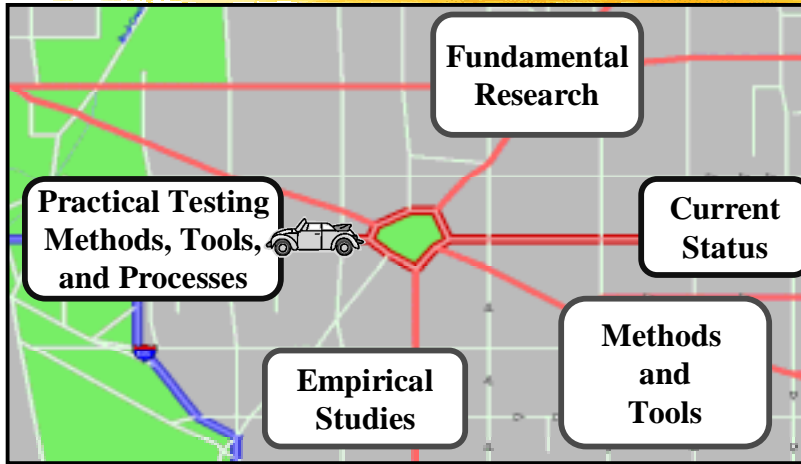
```
Procedure AVG
count = 0
read (n)
while (not EOF) do
  if (n < 0)
    return (error)
  else
    num[count] = n
    count ++
  endif
endwhile
avg = mean (nums, count)
return (avg)
```



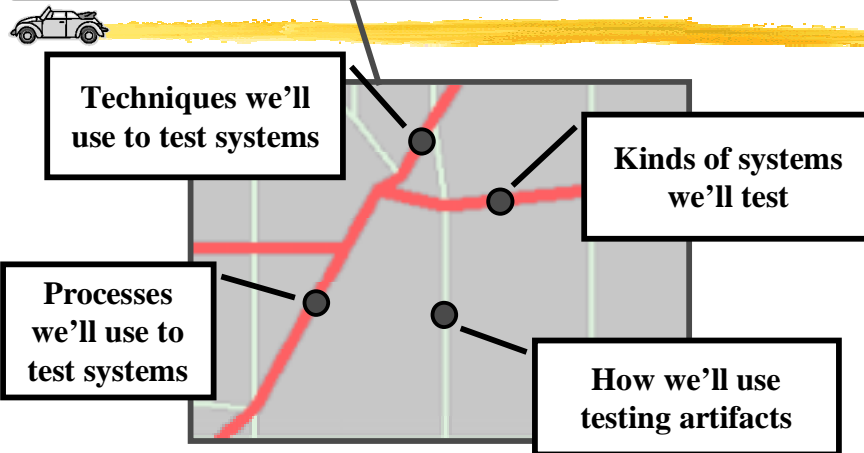
ICSE 2000 Future of Software Engineering

2

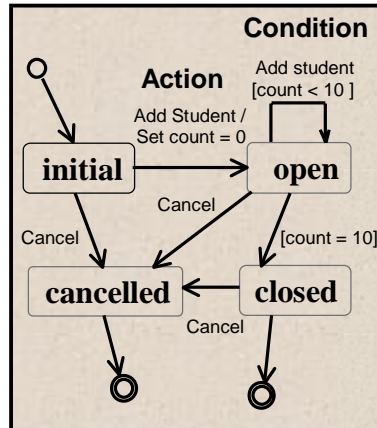
Roadmap



Fundamental Research



Use of Precode Artifacts



- To design test cases
- To assess testability of the software, evaluate alternative architectures to improve testability, etc.
- To improve the testing process



Effectiveness of Testing Techniques



Fault Detection

- Relate classes of faults to testing criteria
- Analyze and empirically evaluate testing criteria
- Combine various testing criteria



Efficiency

- Evaluate practicality of testing criteria
- Develop demand, incremental, or partial instrumentation and evaluation techniques
- Approximate complete coverage



Component-Based Systems



Component Provider



- Tests independently of the application
- Evaluates various qualities
- Identifies, information required by component user

Component User



- Tests in context of the application
- Integrates components w/application
- Extracts and tests parts of component used by application



ICSE 2000 Future of Software Engineering

7

Evolving Systems



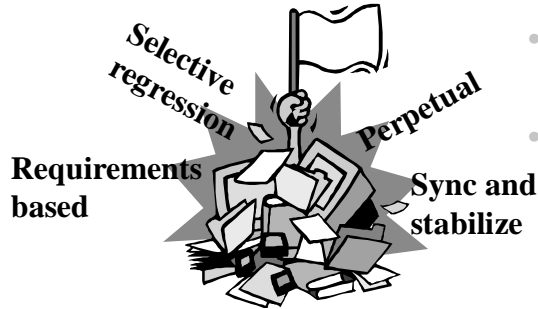
- Systems evolve because of
 - maintenance (bugs, changing requirements, etc.)
 - lack of, or inadequate, requirements
- Need
 - selective retest for modified software
 - prediction models use precode and testing artifacts



ICSE 2000 Future of Software Engineering

8

Create Effective Testing Processes



- Validate existing models
- Integrate testing w/ development and maintenance processes

- Combine testing with other quality processes or analyses



Use Testing Artifacts



- Infer various aspects of program behavior
- Provide developers with information appropriate to interests
- Influence the guidelines for development of the software

Execution Traces, Prog State	Coverage Information	Performance Statistics
	<pre> Procedure AVG count = 0 read (n) while (not EOF) do if (n < 0) return (error) else num[count] = n count ++ endif endwhile avg = mean (nums, count) return (avg) </pre>	
	Pass/Fail Results	

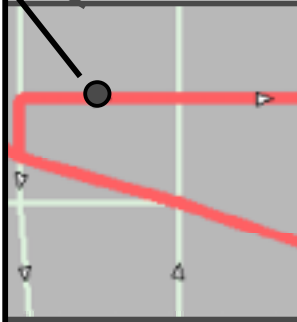


Methods and Tools



Technology transfer requires demonstration of effectiveness, in practice, of the techniques

Demonstration of effectiveness requires robust prototypes of the techniques for use in experimentation



Methods and Tools



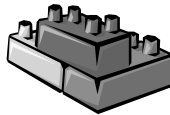
Scalable to large system

Process real code, real systems; probably integrate into industrial environment



Usable by practitioners

Produce useful reports so that practitioners can see benefits

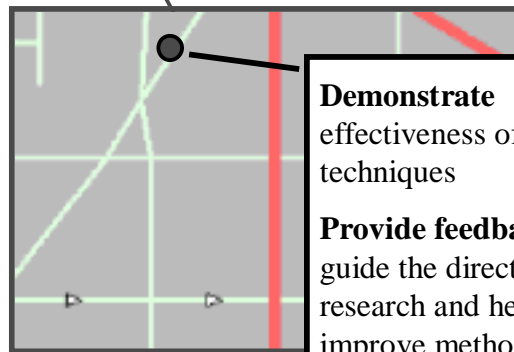


Buildable with existing infrastructure

Need available infrastructure for building prototypes; probably from industry



Empirical Studies



Demonstrate
effectiveness of
techniques

Provide feedback to
guide the direction of
research and help
improve methods and
tools



Empirical Studies



Need software subjects for use in
evaluation of efficiency and
effectiveness of techniques



Need human subjects for use in
evaluation of the usability of the
prototype tools

Need the support of industry to make subjects
available and provide resources to enable studies



Conclusion



- Quality of software will become an essential component of a product's success
- Key areas for testing research will be
 - Fundamental research
 - Methods and tools
 - Empirical evaluation
- Testing will continue to be a major method for ensuring quality; thus research will continue