Tester-Centric Automated Testing: Bringing Humans Into the Loop

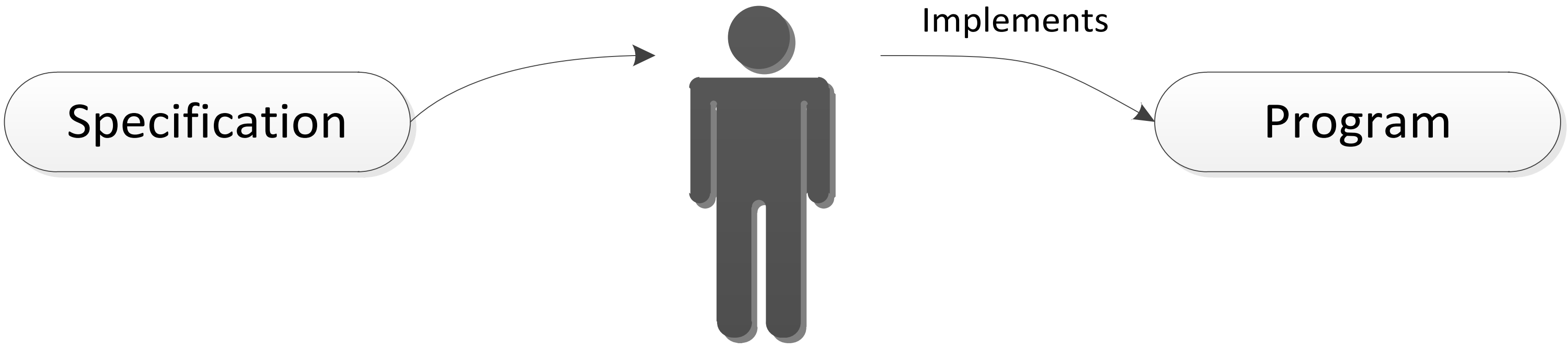
Matt Staats

KAIST

Outline

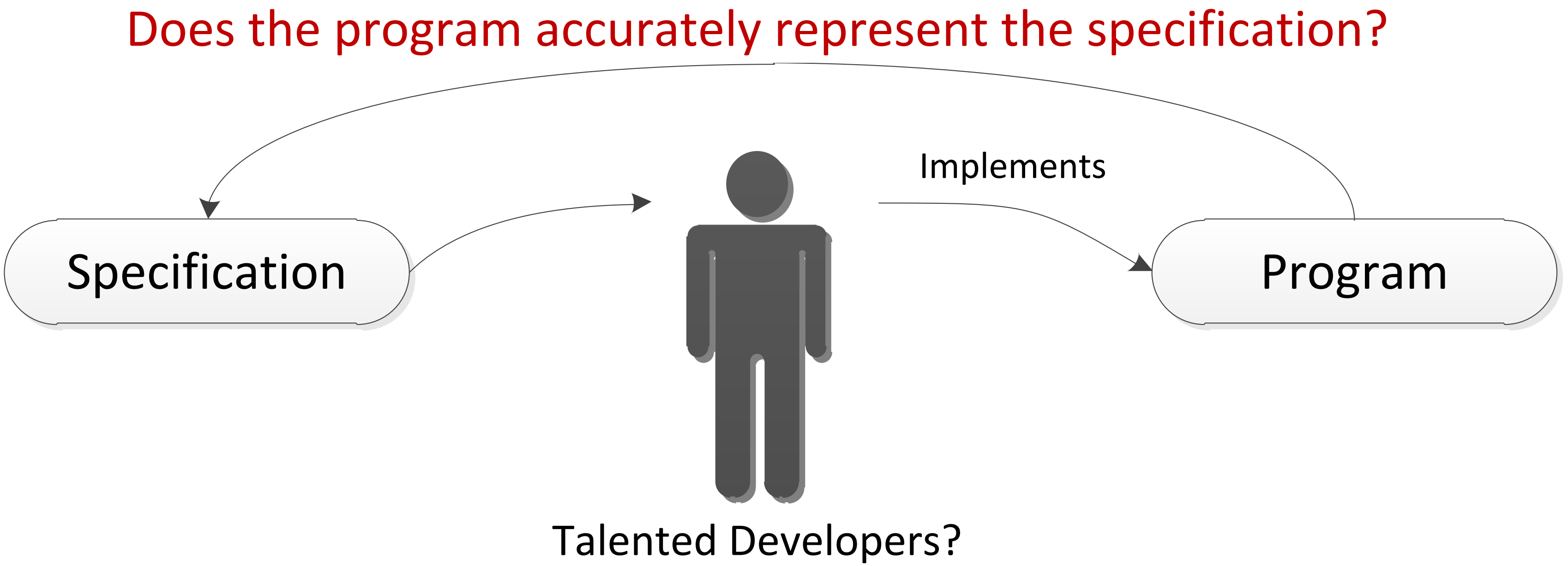
- Overview of software testing
- Highlight the problem
- Review of recent work toward solution
  - Empirical study demonstrating alternative approach is difficult to use
  - Application of technique demonstrating potential value
- Future work centered on concept of “tester-centric automated testing”

Software Development

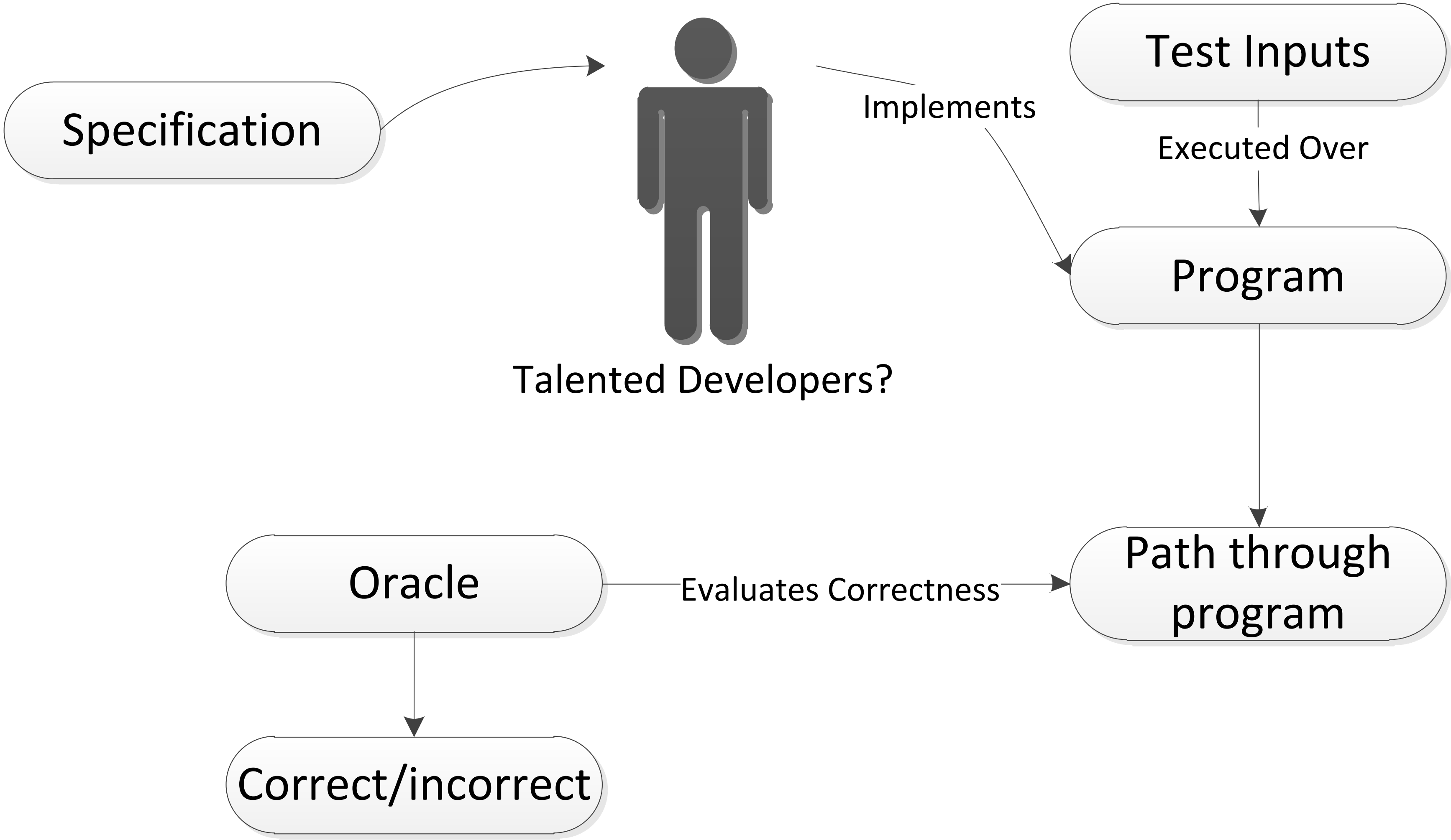


Talented Developers?

The Big Question



Testing Process


Specification

Talented Developers?

Implements

Test Inputs

Executed Over

Program

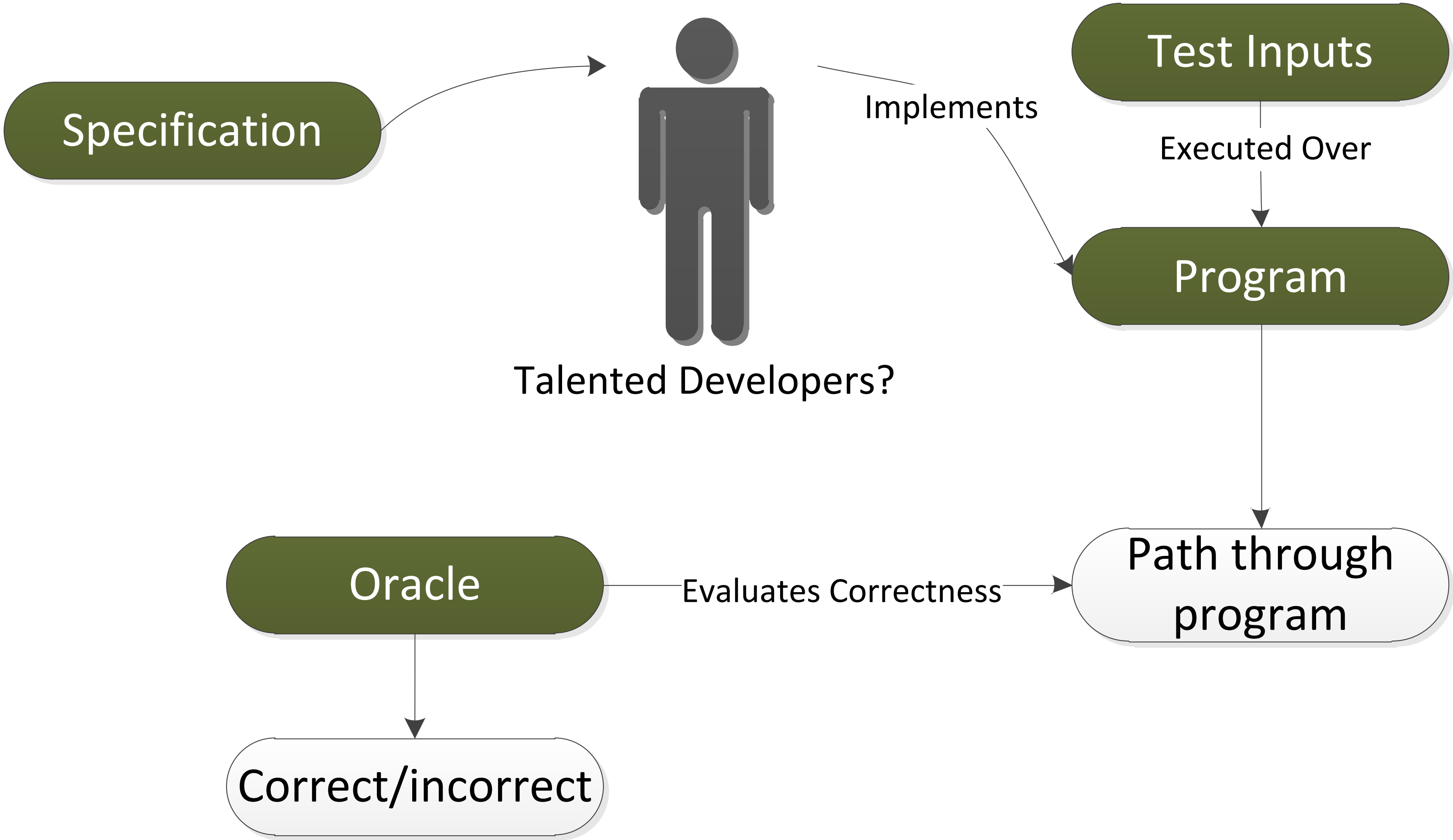
Path through program

Oracle

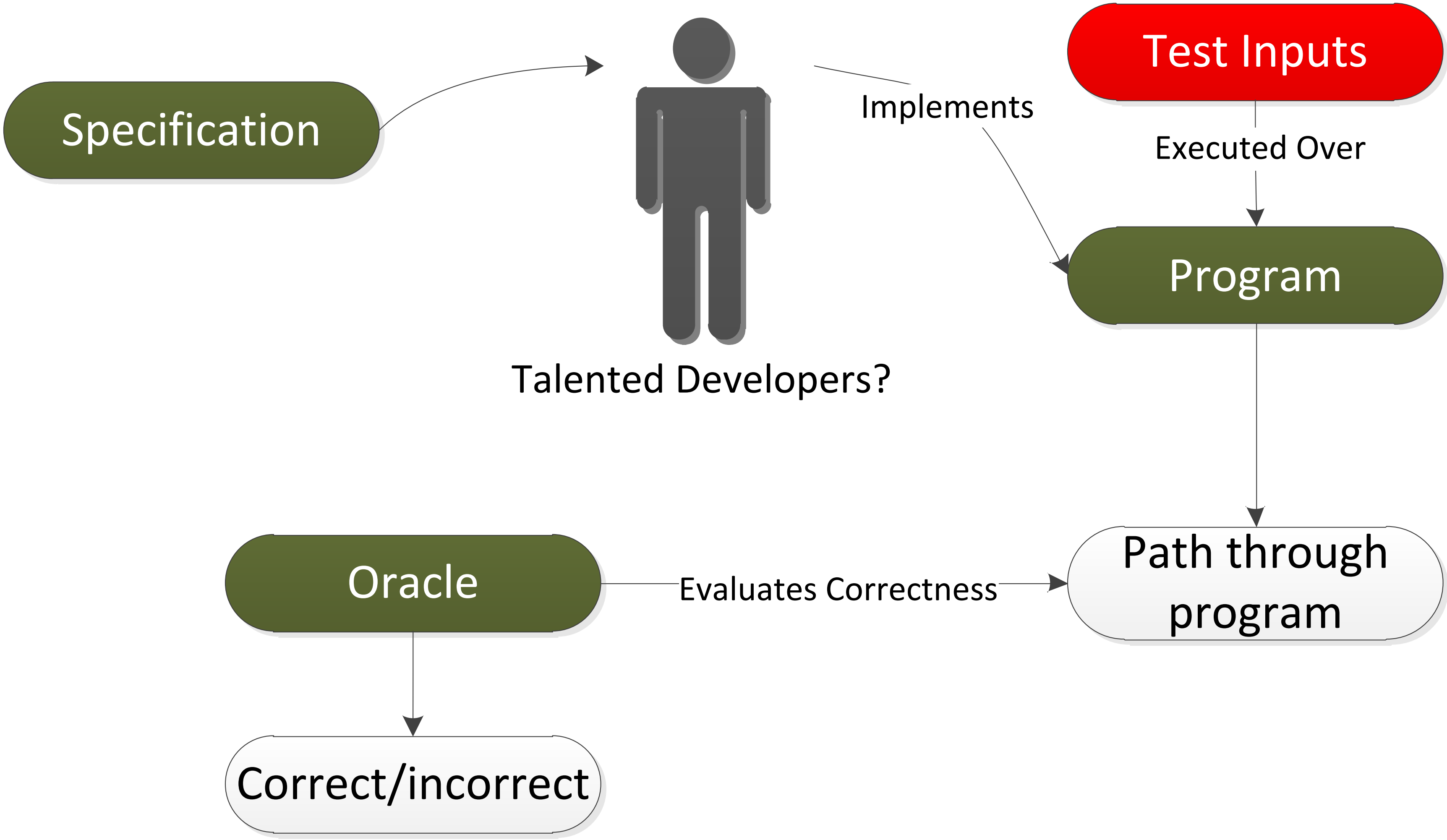
Evaluates Correctness

Correct/incorrect

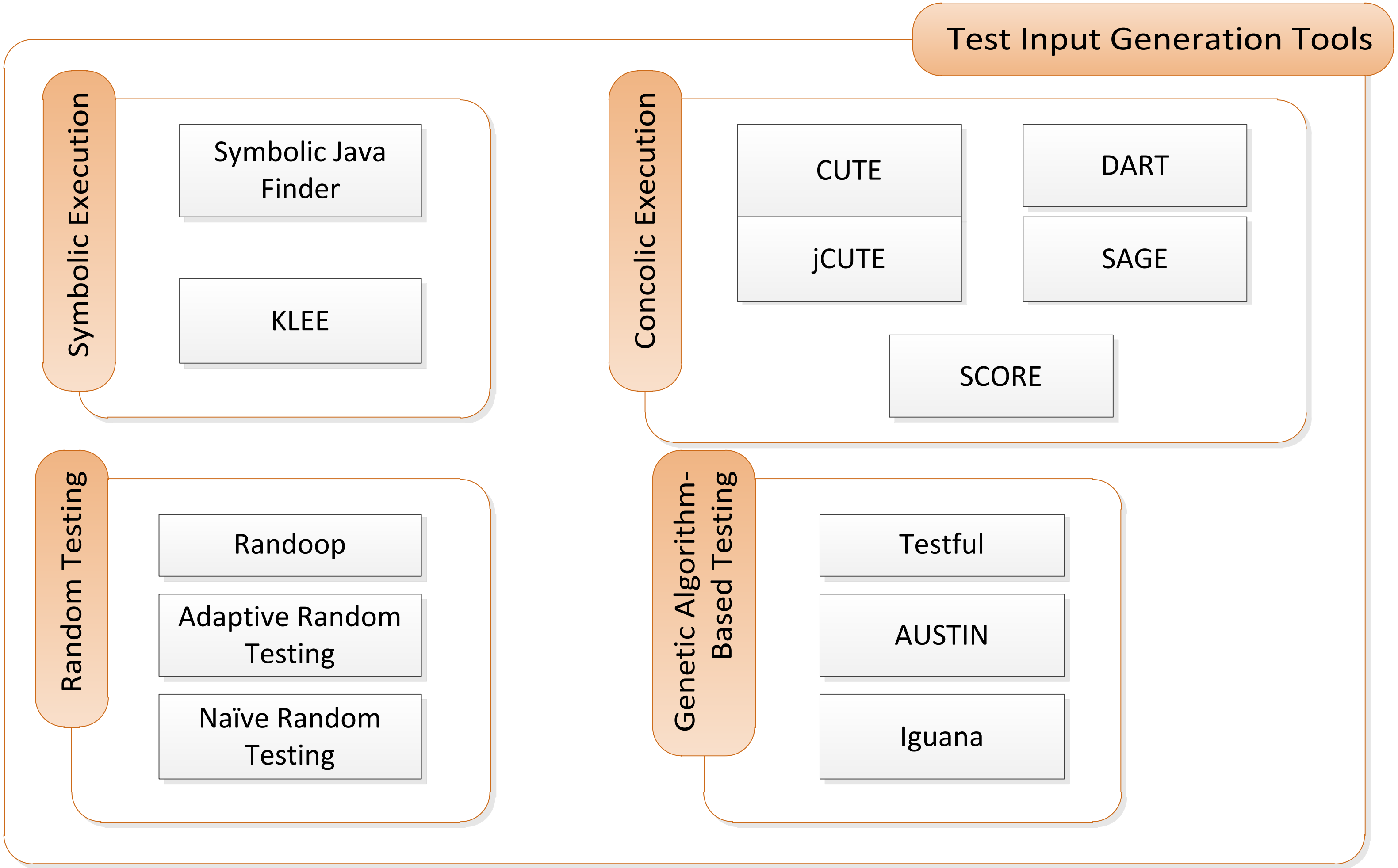
Testing Process



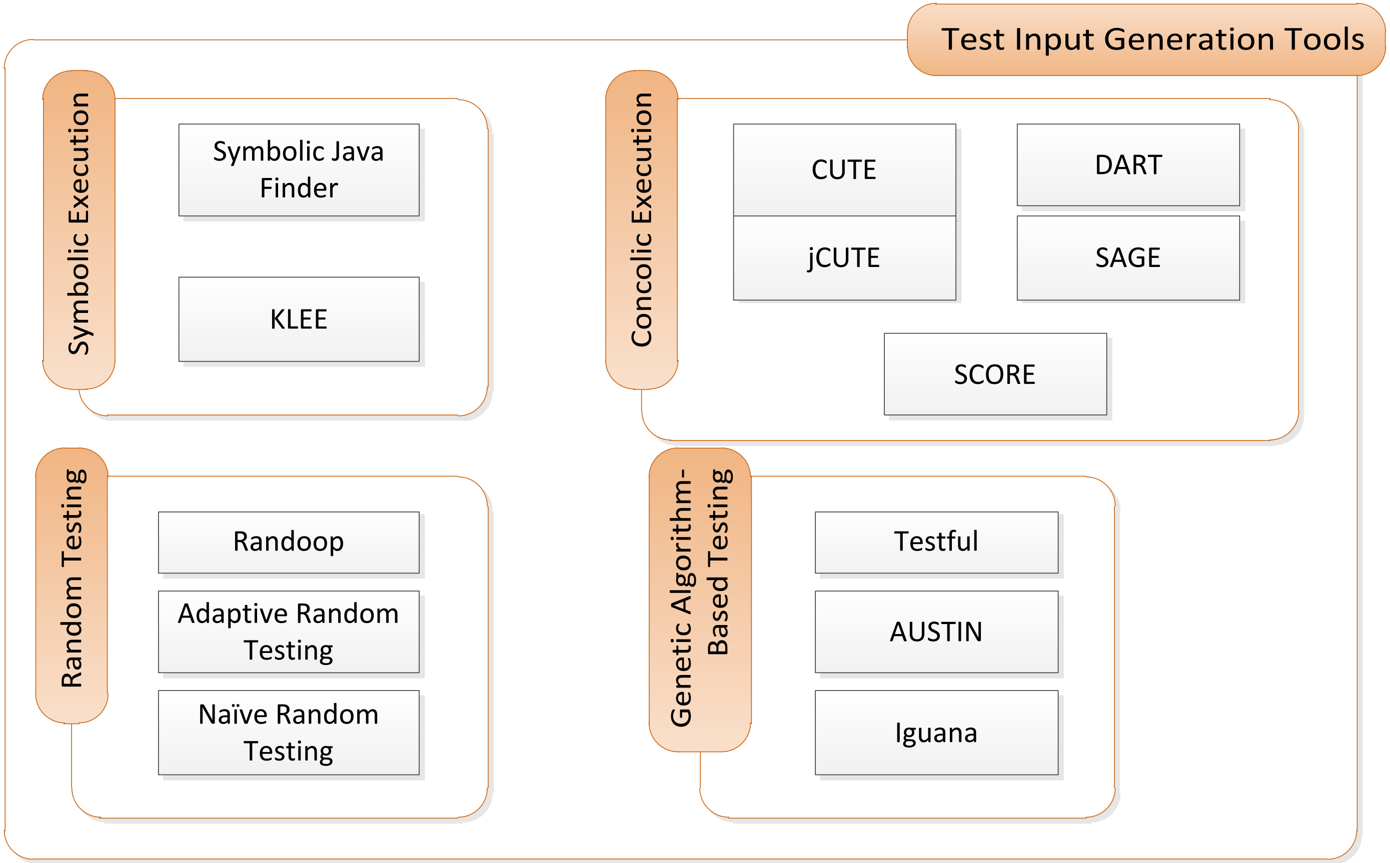
Testing Artifacts – In Practice



Test Input Generation



Test Input Generation



Test Input Generation Tools

Symbolic Execution

- Symbolic Java Finder
- KLEE

Concolic Execution

- CUTE
- jCUTE
- DART
- SAGE
- SCORE

Random Testing

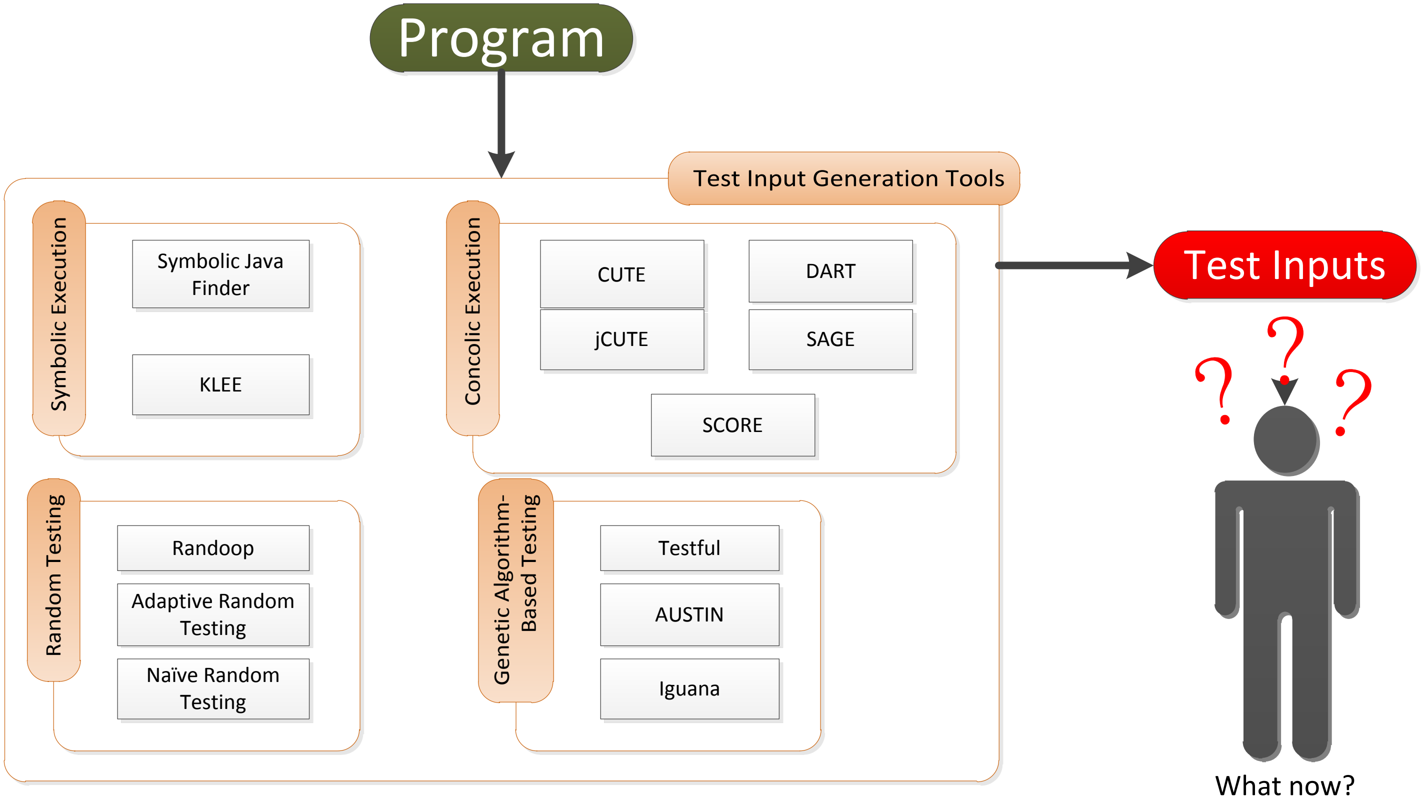
- Randoop
- Adaptive Random Testing
- Naïve Random Testing

Genetic Algorithm-Based Testing

- Testful
- AUSTIN
- Iguana

- Good work, progress in reachability, efficiency, etc.

Test Input Generation



Program

Test Input Generation Tools

Symbolic Execution

Symbolic Java Finder

KLEE

Concolic Execution

CUTE

DART

jCUTE

SAGE

SCORE

Random Testing

Randoop

Adaptive Random Testing

Naïve Random Testing

Genetic Algorithm-Based Testing

Testful

AUSTIN

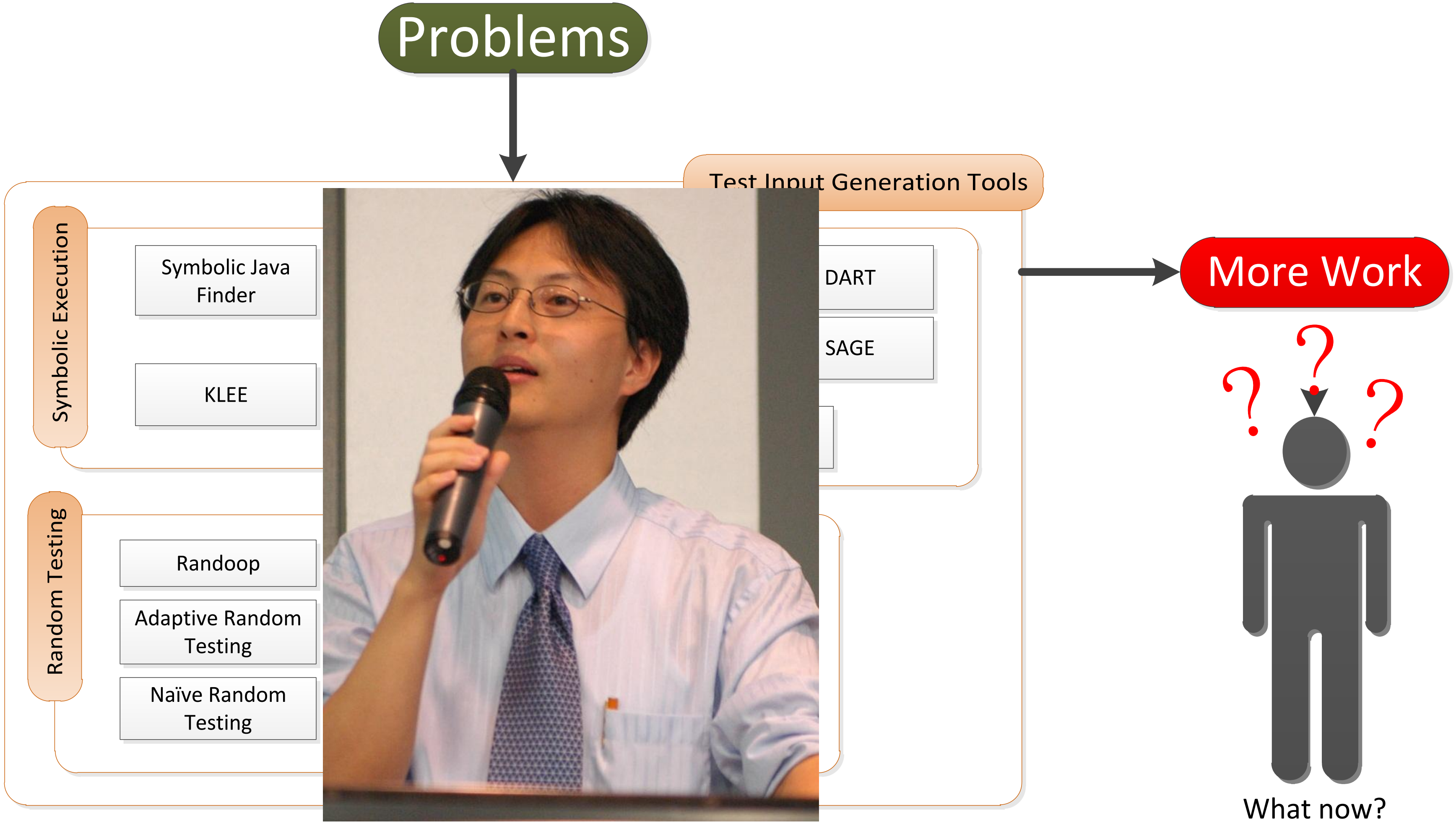
Iguana

Test Inputs

What now?

- Unclear how users can use tools
- We make *a lot* of work for people

Test Input Generation



Problems

Test Input Generation Tools

Symbolic Execution

Symbolic Java Finder

KLEE

DART

SAGE

Random Testing

Randoop

Adaptive Random Testing

Naïve Random Testing

More Work

What now?

Test Input Generation

Problems

Test Input Generation Tools

Symbolic Execution

- Symbolic Java Finder
- KLEE

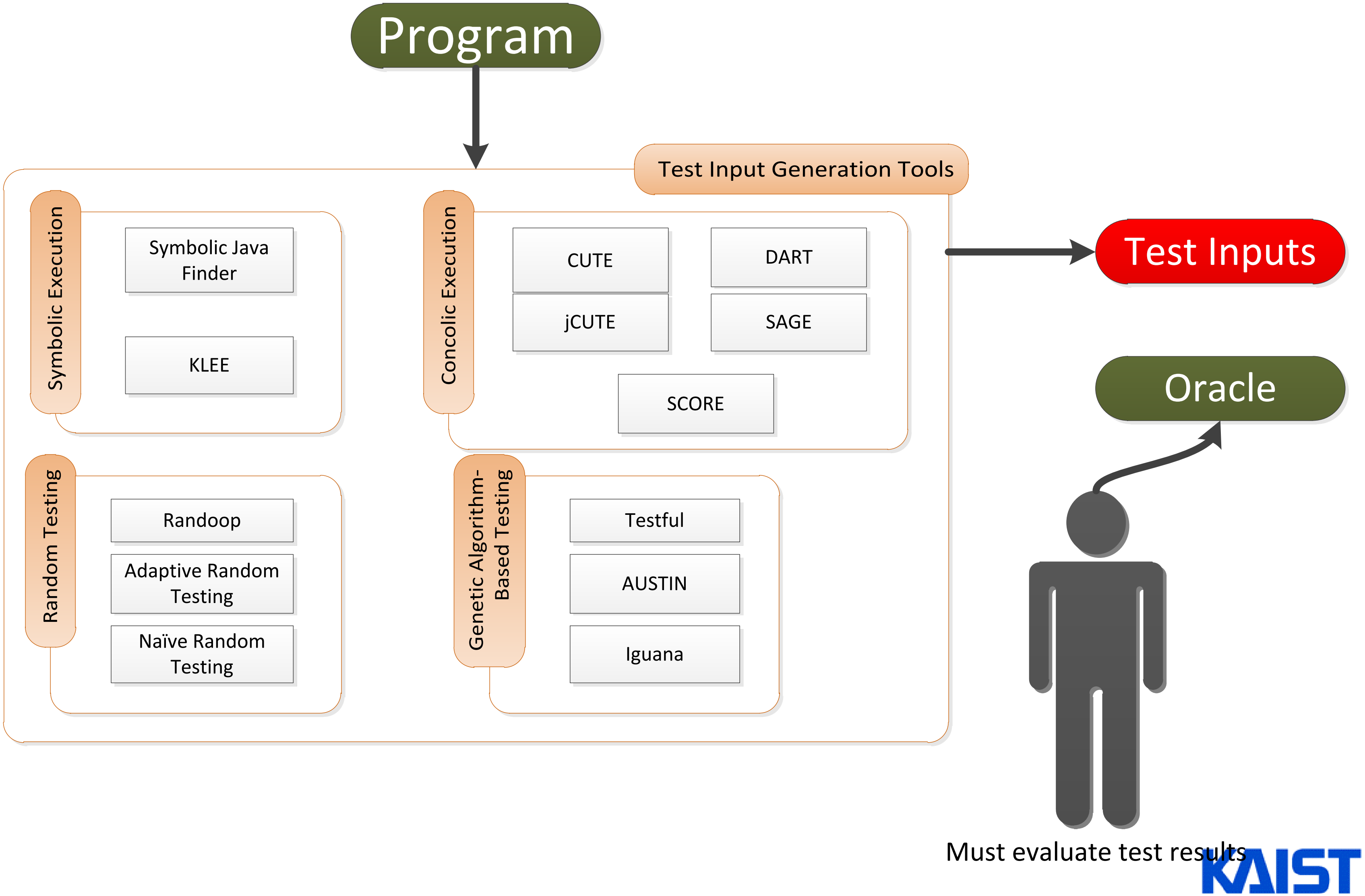
Random Testing

- Randoop
- Adaptive Random Testing
- Naïve Random Testing

- DART
- SAGE

More Work

Problem: No Support for Test Oracles



Program

Test Input Generation Tools

Symbolic Execution

Symbolic Java Finder

KLEE

Concolic Execution

CUTE

DART

jCUTE

SAGE

SCORE

Random Testing

Randoop

Adaptive Random Testing

Naïve Random Testing

Genetic Algorithm-Based Testing

Testful

AUSTIN

Iguana

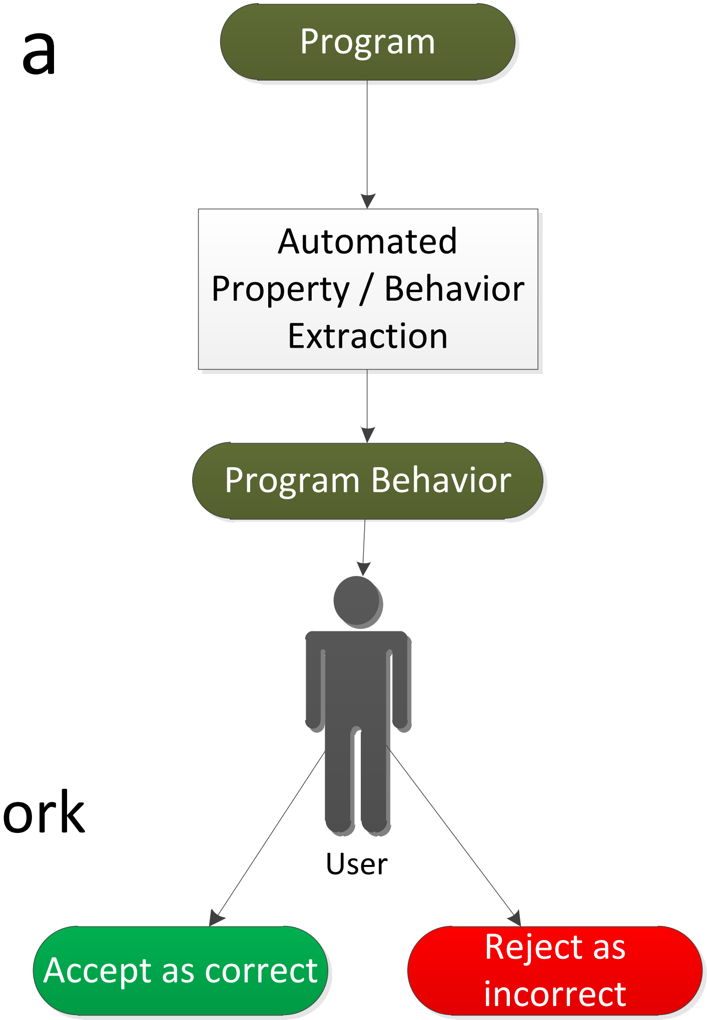
Test Inputs

Oracle

Must evaluate test results

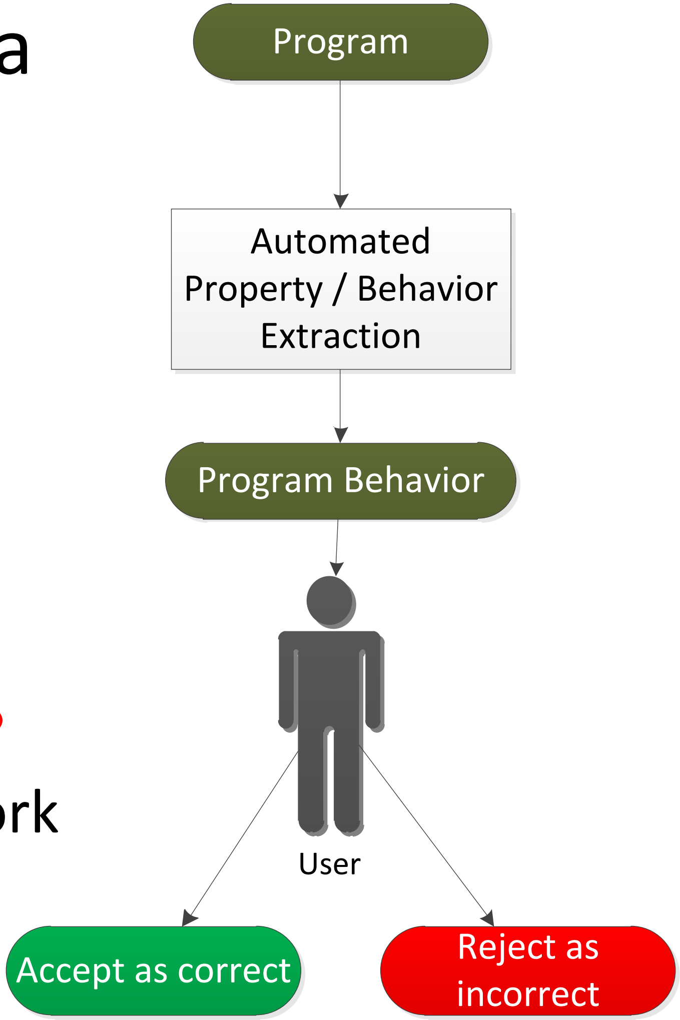
Existing Work: Automatic Oracle Generation

- Idea: automatically generate a test oracle from the system
- User then (necessarily) evaluates result
- Several approaches, varying result
  - – Program invariant generation
    - *Daikon*, *AutoInfer*, Xie/Notkin work
  - – Trace generation
    - EvoSuite

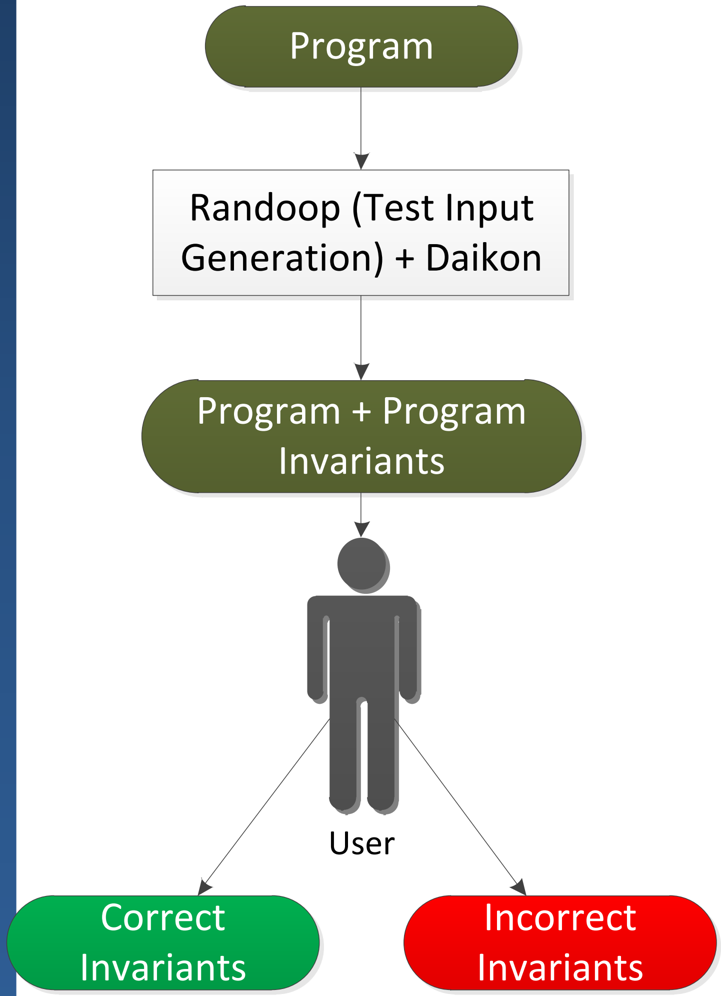


Existing Work: Automatic Oracle Generation

- Idea: automatically generate a test oracle from the system
- User then (necessarily) evaluates result
- Several approaches, varying result
  - Program invariant generation
    - *Daikon*, *AutoInfer*, Xie/Notkin work
  - Trace generation
    - EvoSuite

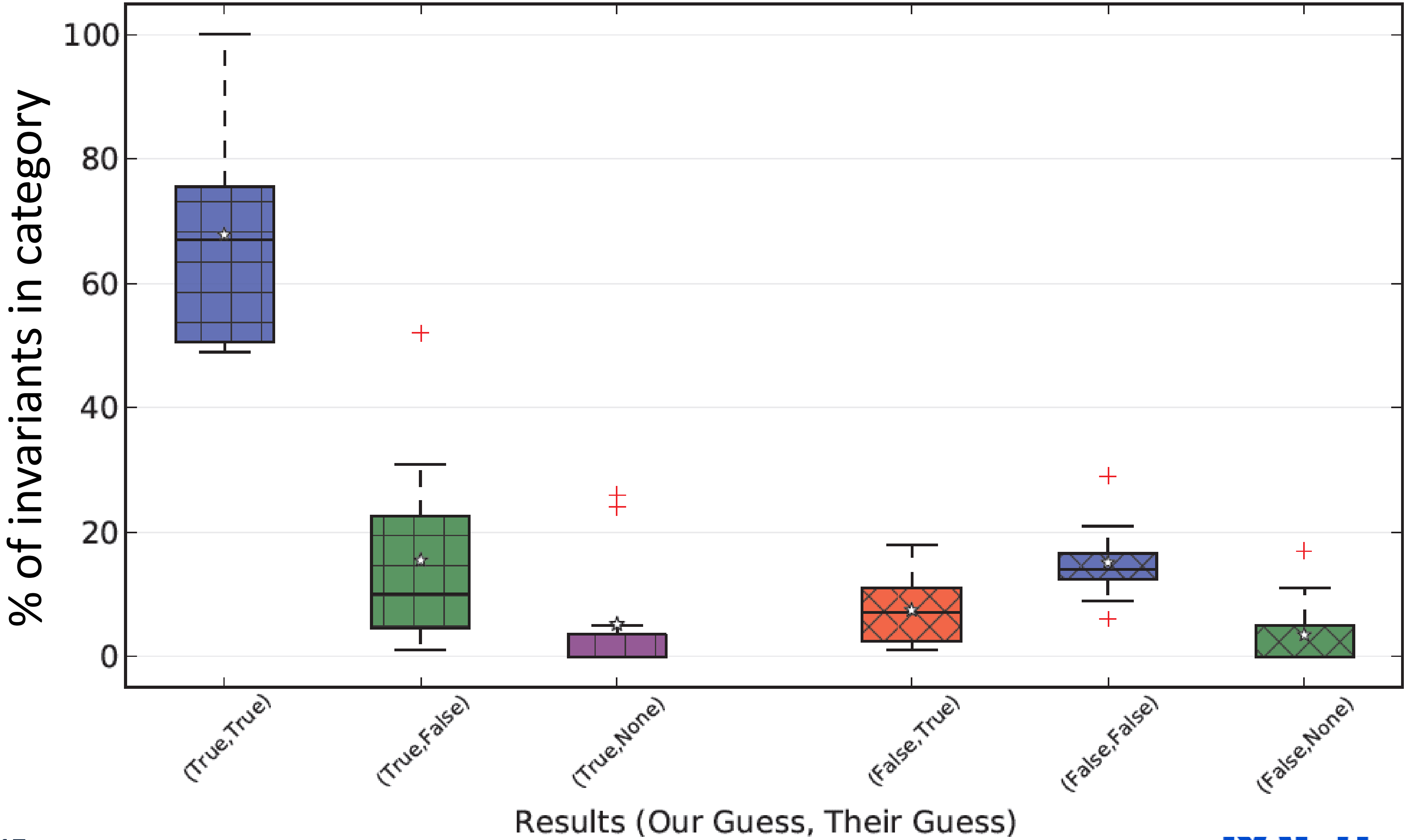


Automatic Invariant Generation

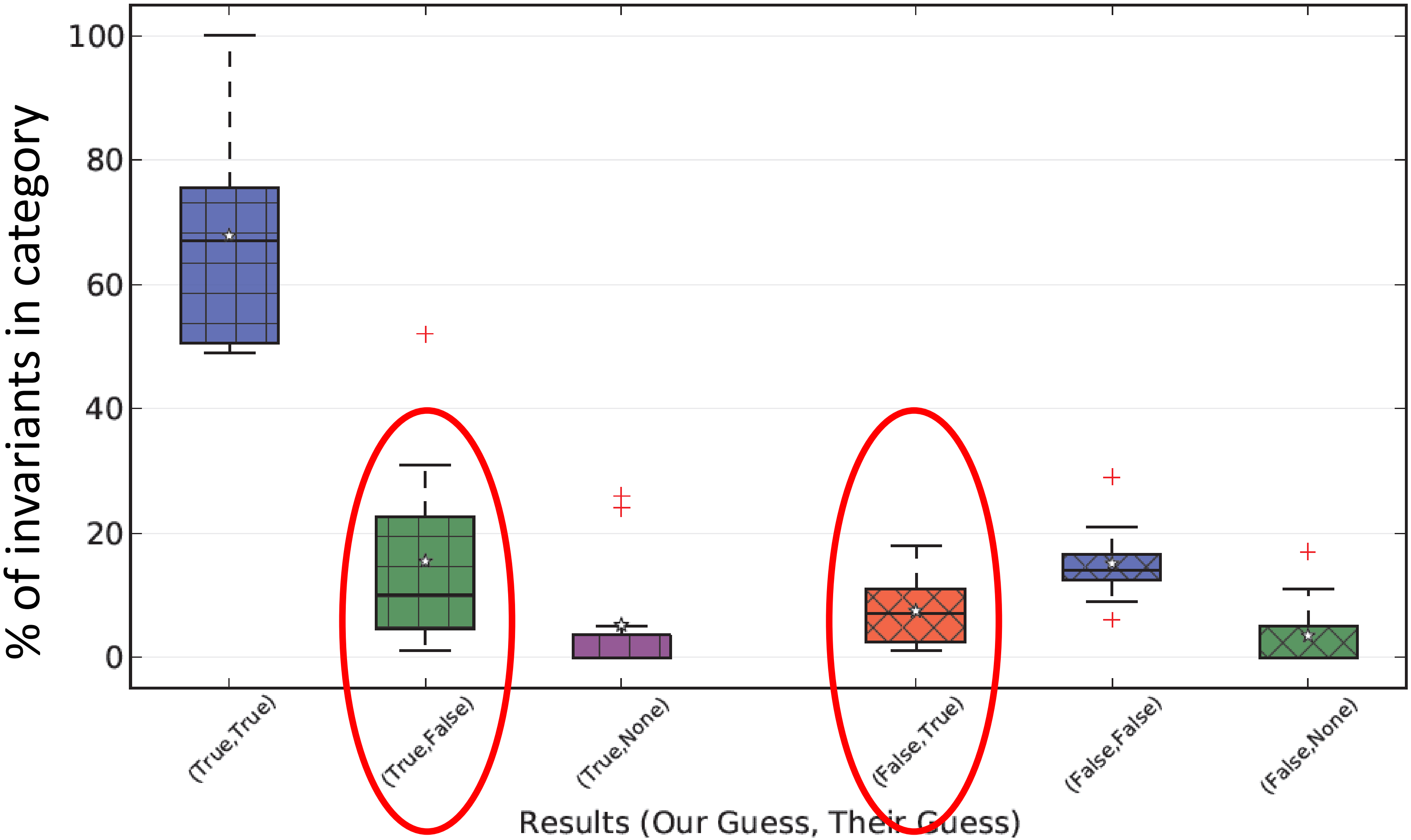

Program
Randoop (Test Input Generation) + Daikon
Program + Program Invariants
User
Correct Invariants
Incorrect Invariants

- Represents potential solution for supporting oracle creation
- Unclear how effective users are at classifying results
  - Problems if poor
  - Little evidence in favor of use
- Study: Daikon dynamic invariant generator
  - Approx. 30 students
  - 3 programs
- Thanks to 최윤자, 김문주 for loaning students

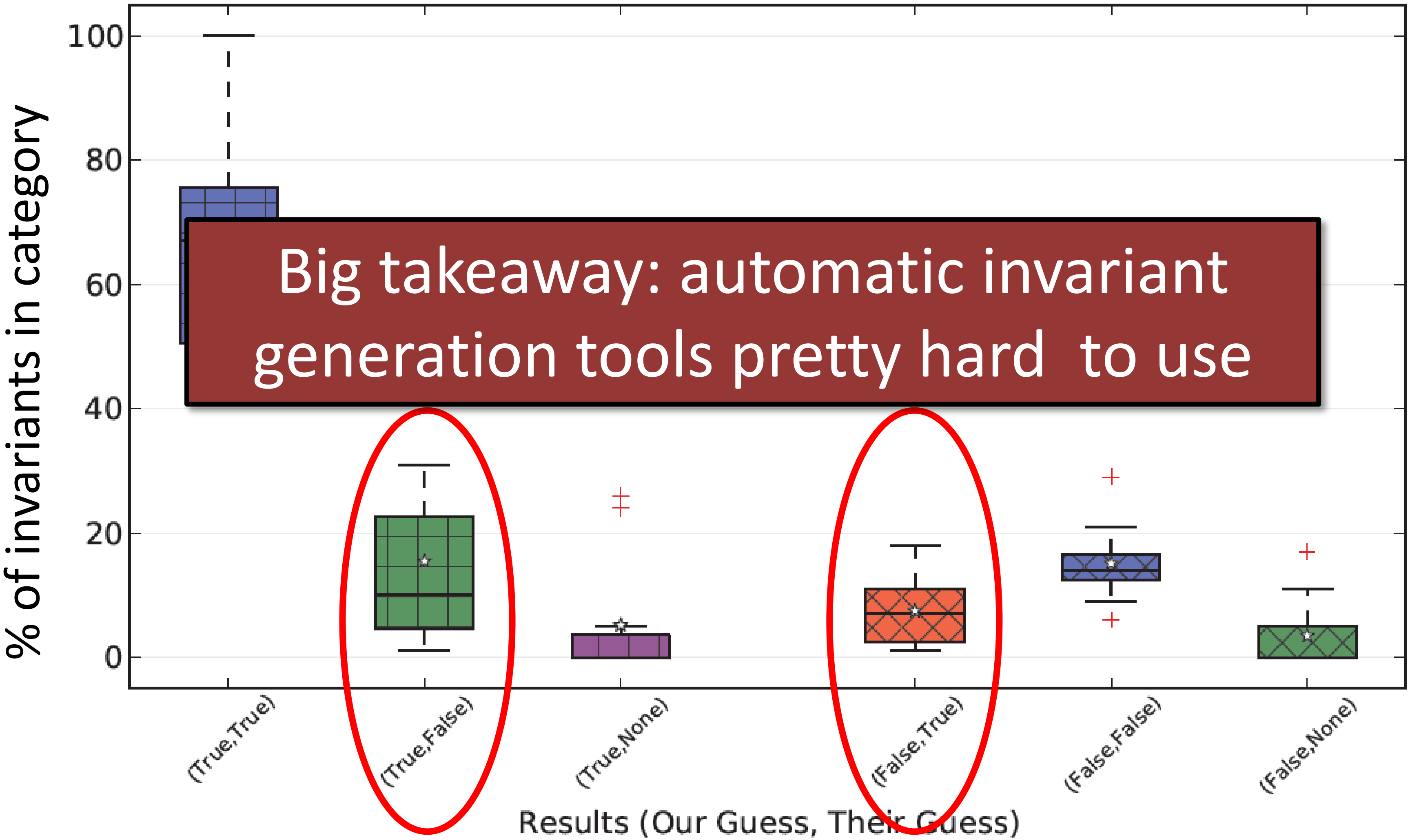
Daikon: User Effectiveness



Daikon: User Effectiveness

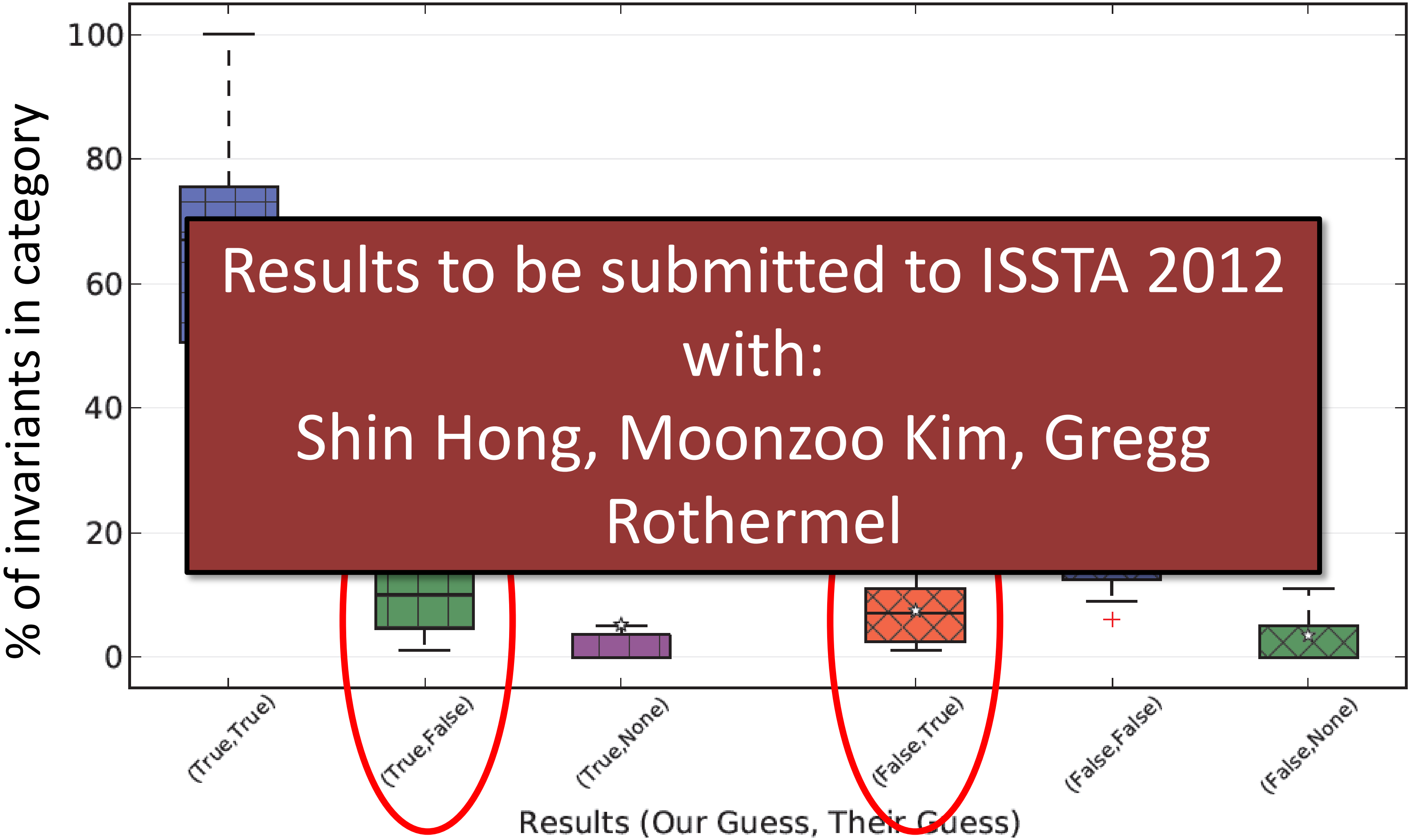


Daikon: User Effectiveness



Big takeaway: automatic invariant generation tools pretty hard to use

Daikon: User Effectiveness

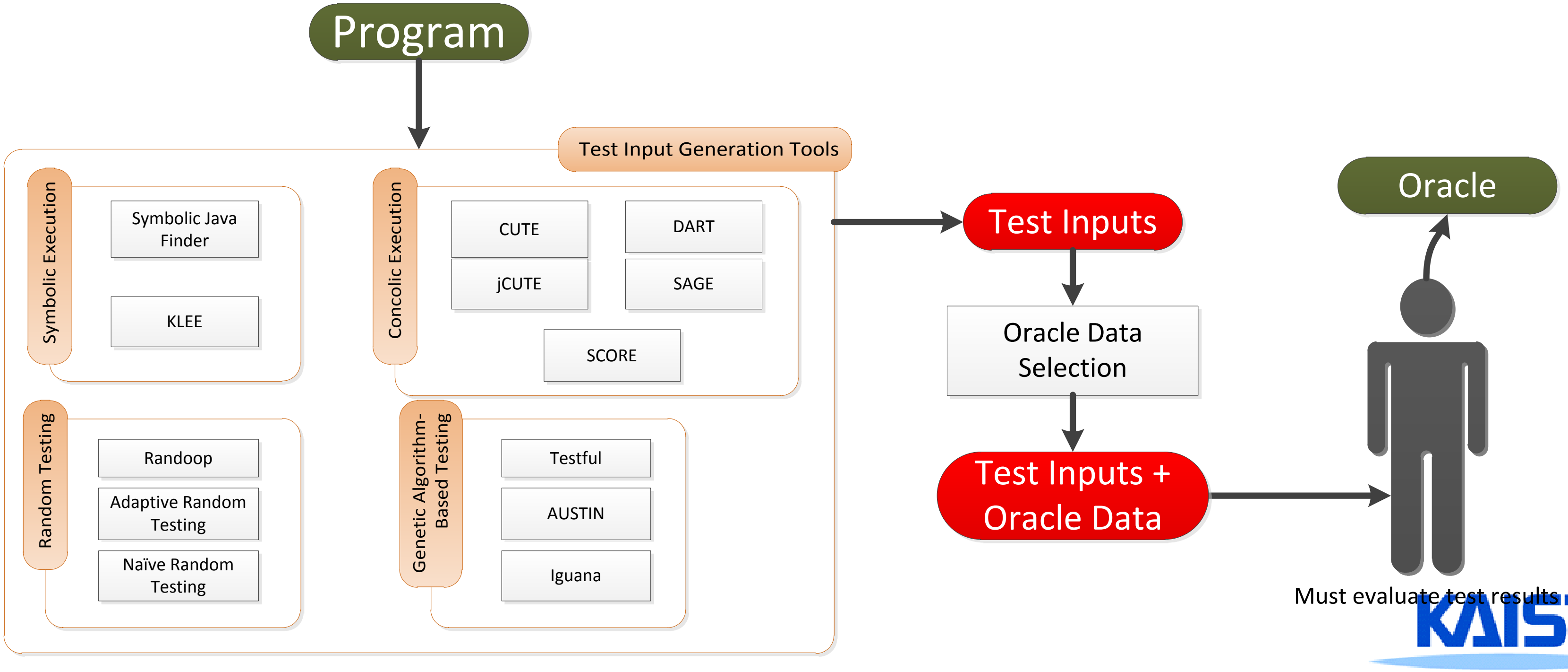


Results to be submitted to ISSTA 2012 with:

Shin Hong, Moonzoo Kim, Gregg Rothermel

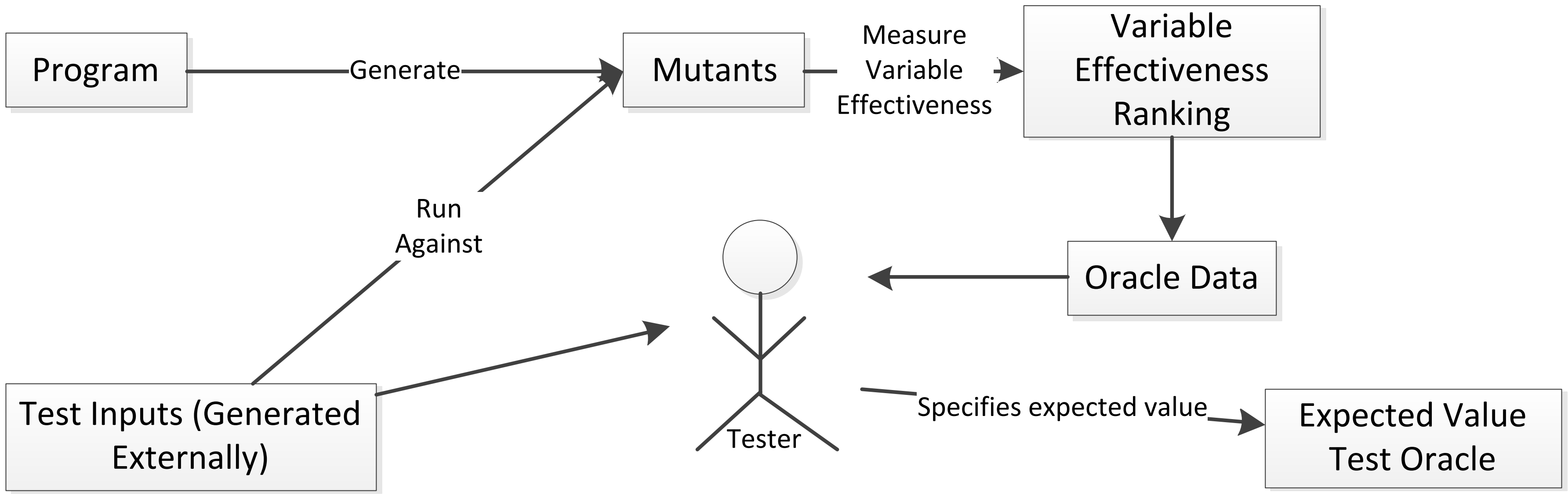
Test Oracle Generation Support

- As an alternative to complete construction, we thought we could support users in making oracles
- Select *oracle data*: part of system oracle defined over
- User still has to define oracle

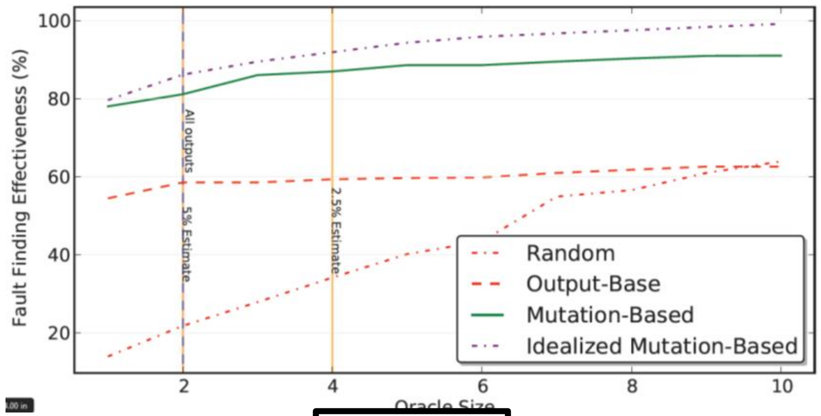


Test Oracle Generation Support

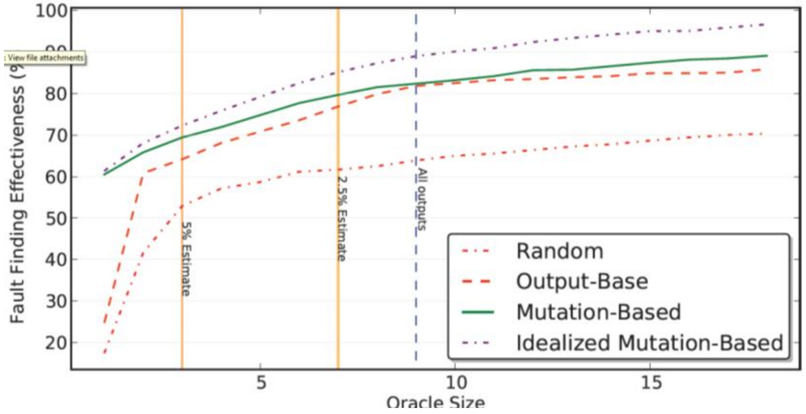
- Mutation testing was used
  - Change program several small ways
  - Determine where and when we can detect changes
- Result is that for a set of test inputs, person has a list of useful variables
- Goal: do better than other methods of selecting oracle data


Program
Generate
Mutants
Measure Variable Effectiveness
Variable Effectiveness Ranking
Run Against
Oracle Data
Test Inputs (Generated Externally)
Tester
Specifies expected value
Expected Value Test Oracle

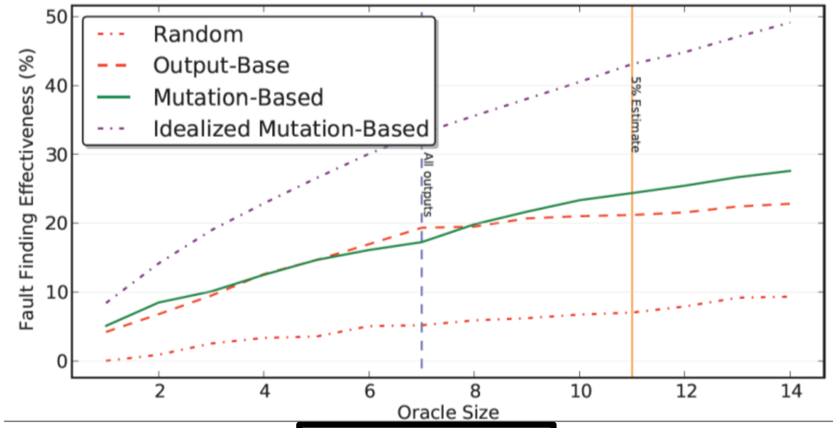
Test Oracle Generation Support



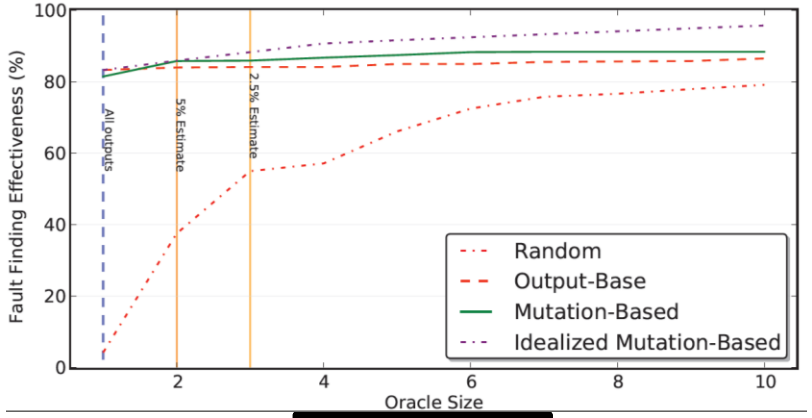
Latctl



Person

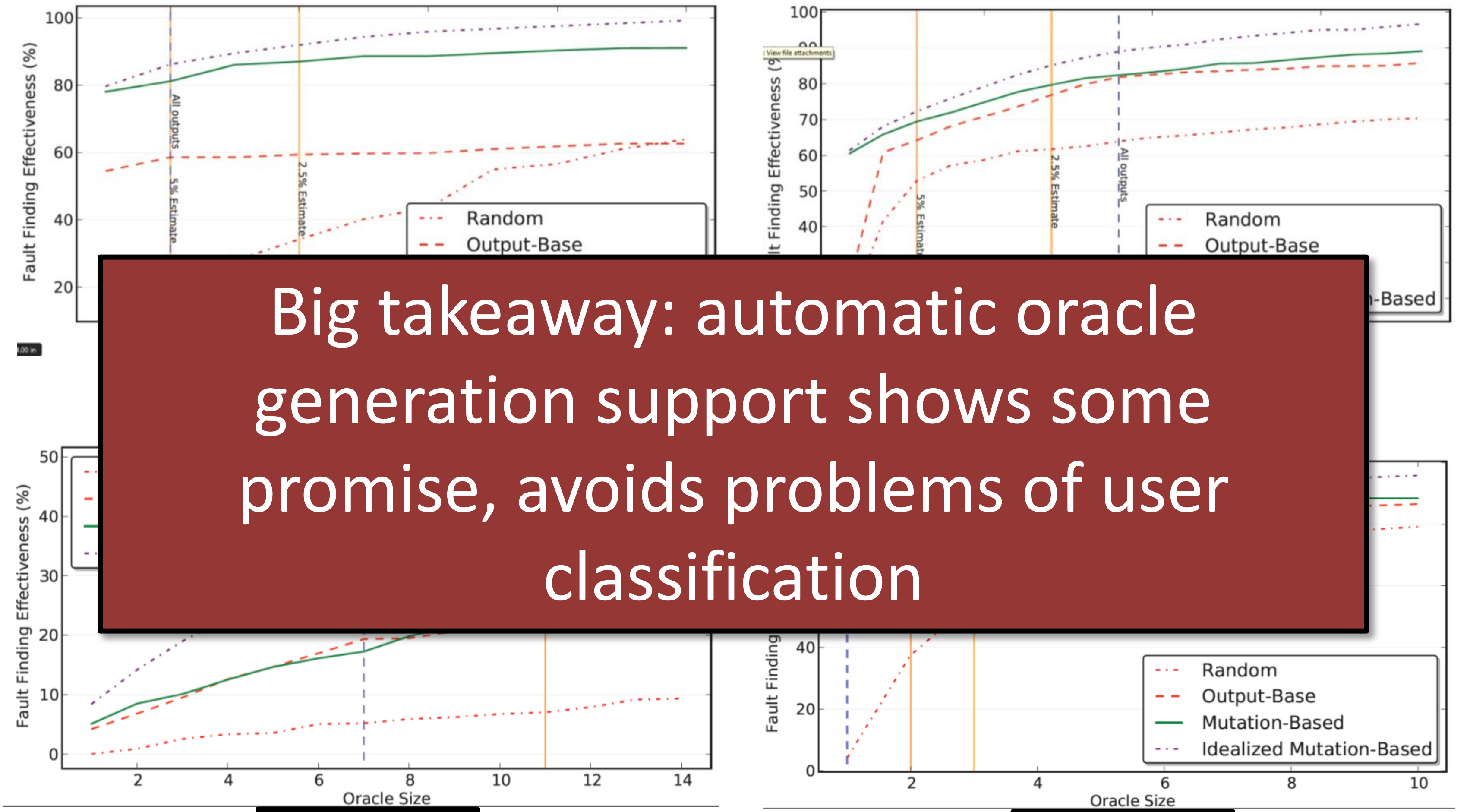


DWM_2



DWM_1

Test Oracle Generation Support

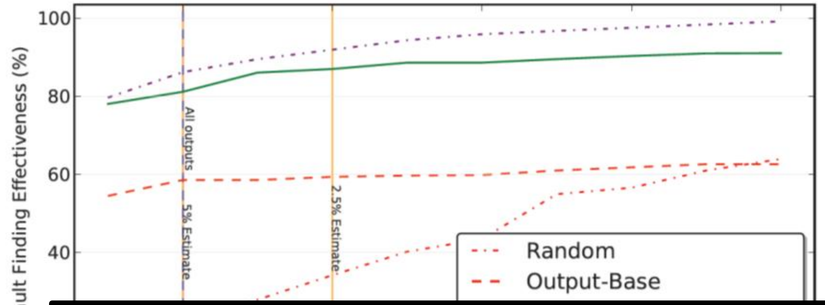


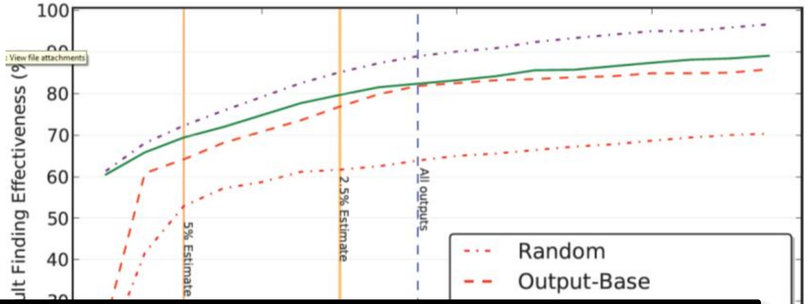
Big takeaway: automatic oracle generation support shows some promise, avoids problems of user classification

DWM_2

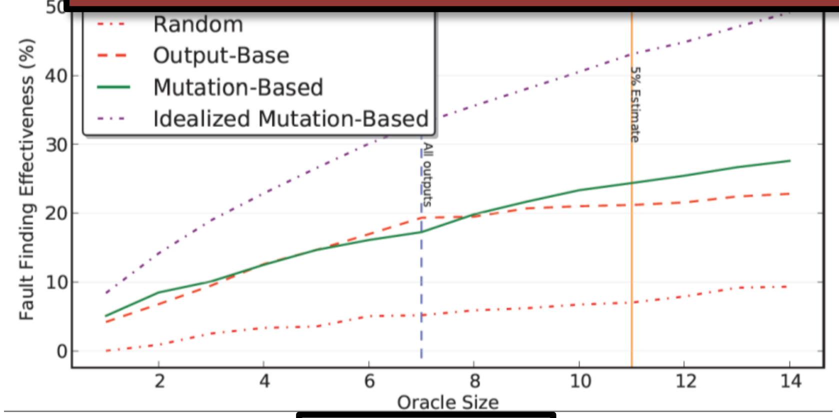
DWM_1

Test Oracle Generation Support

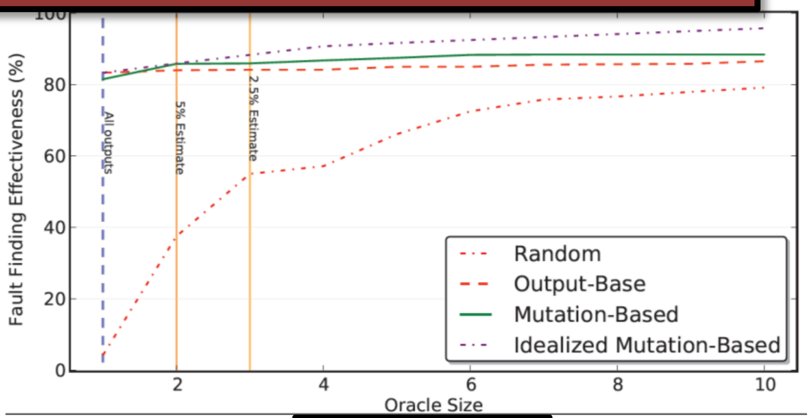




Results under submission to ICSE 2012 with:
Greg Gay, Mats Heimdahl



DWM_2



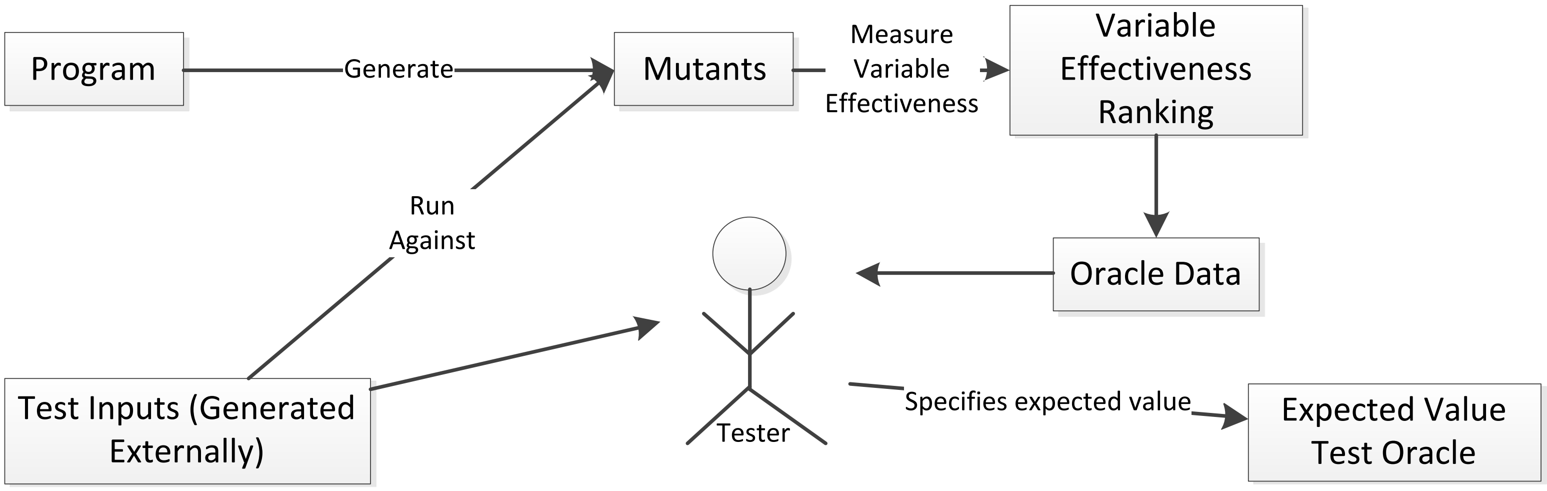
DWM_1

Tester-Centric Automated Testing

- High-level problem: poor integration of users into automated testing techniques
  - Current techniques are very (maybe too) demanding on users
  - Our own approach provides direction, has promise
- Three takeaways
  - Users are necessary, but often ignored in automated testing
  - Existing methods of supporting users in test oracles have problems
  - Proposed method maybe can do better
- ***Can we do better?***

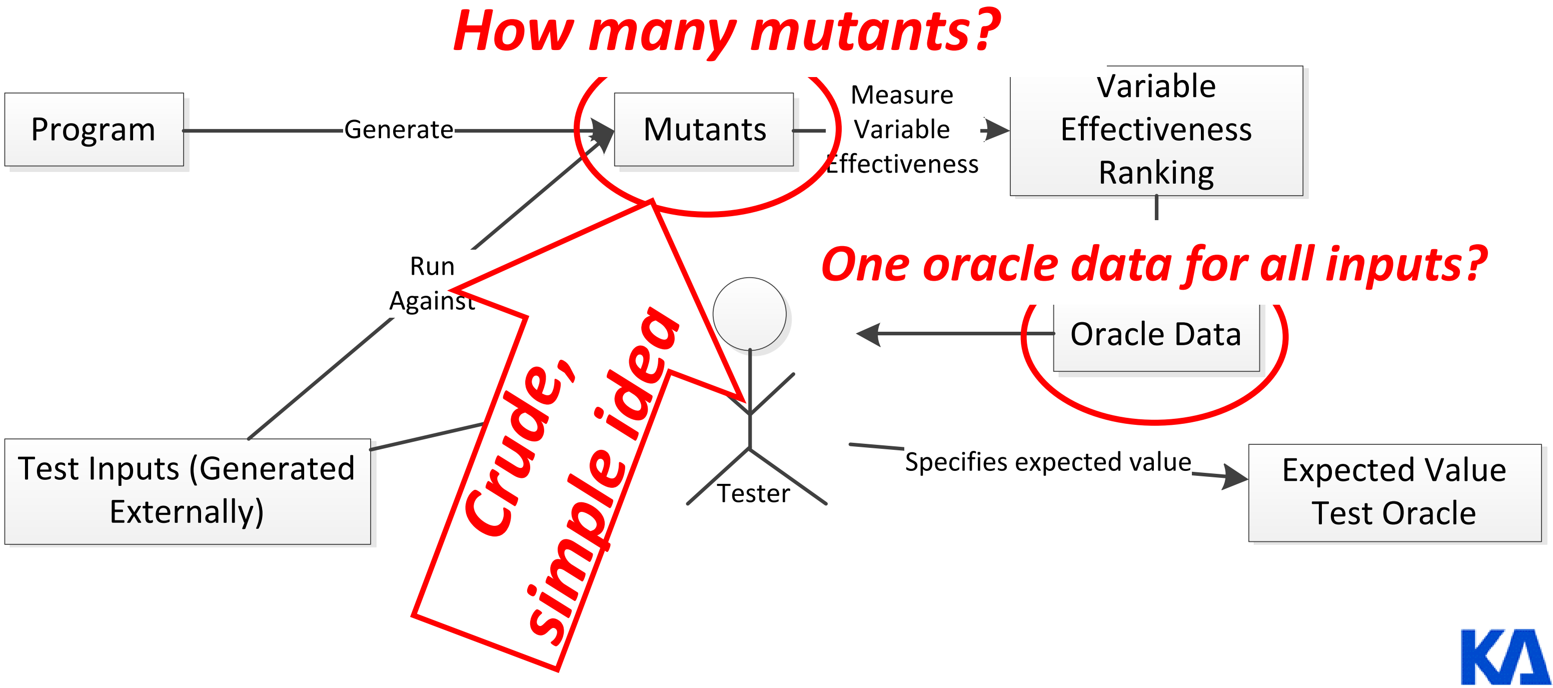
Tester-Centric Automated Testing

- Leads to ideas for future work
- Several problems/issues left
  - **Method of supporting oracle selection is coarse at best**


Program
Generate
Mutants
Measure Variable Effectiveness
Variable Effectiveness Ranking
Run Against
Oracle Data
Tester
Specifies expected value
Test Inputs (Generated Externally)
Expected Value Test Oracle

Tester-Centric Automated Testing

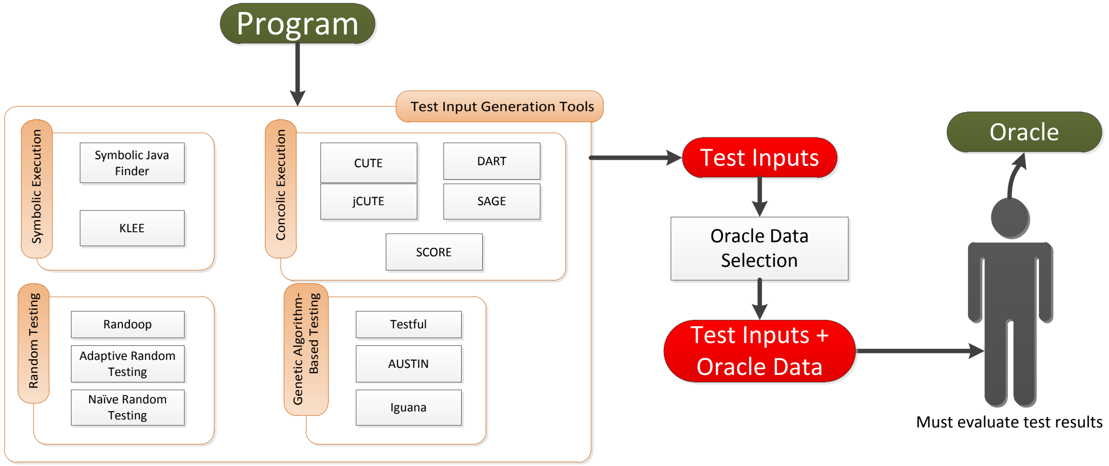
- Leads to ideas for future work
- Several problems/issues left
  - **Method of supporting oracle selection is coarse at best**

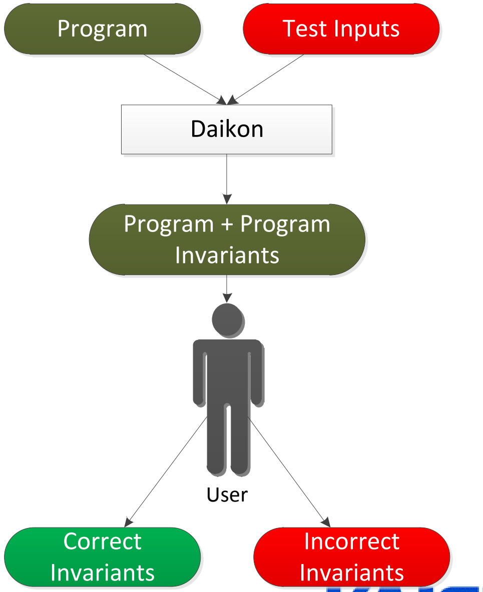


How many mutants?
Program
Generate
Mutants
Measure Variable Effectiveness
Variable Effectiveness Ranking
Run Against
One oracle data for all inputs?
Oracle Data
Crude, simple idea
Tester
Test Inputs (Generated Externally)
Specifies expected value
Expected Value Test Oracle

Tester-Centric Automated Testing

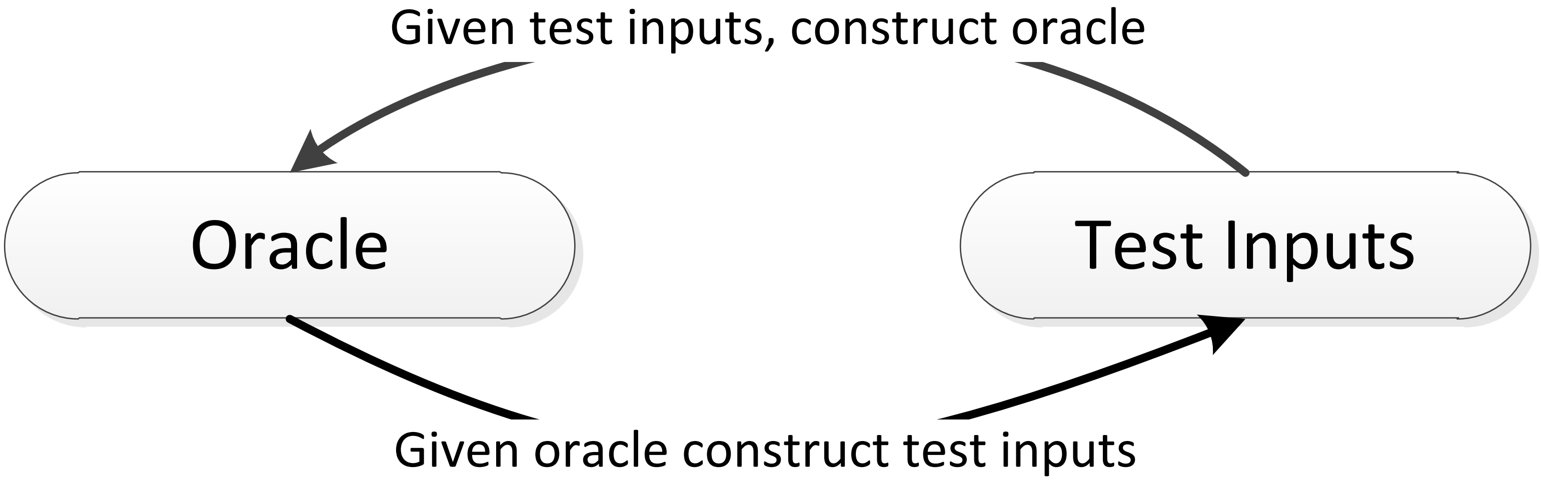
- Leads to ideas for future work
- Several problems/issues left
  - Method of supporting oracle selection is coarse at best
  - **Test input and oracle generation always separate**





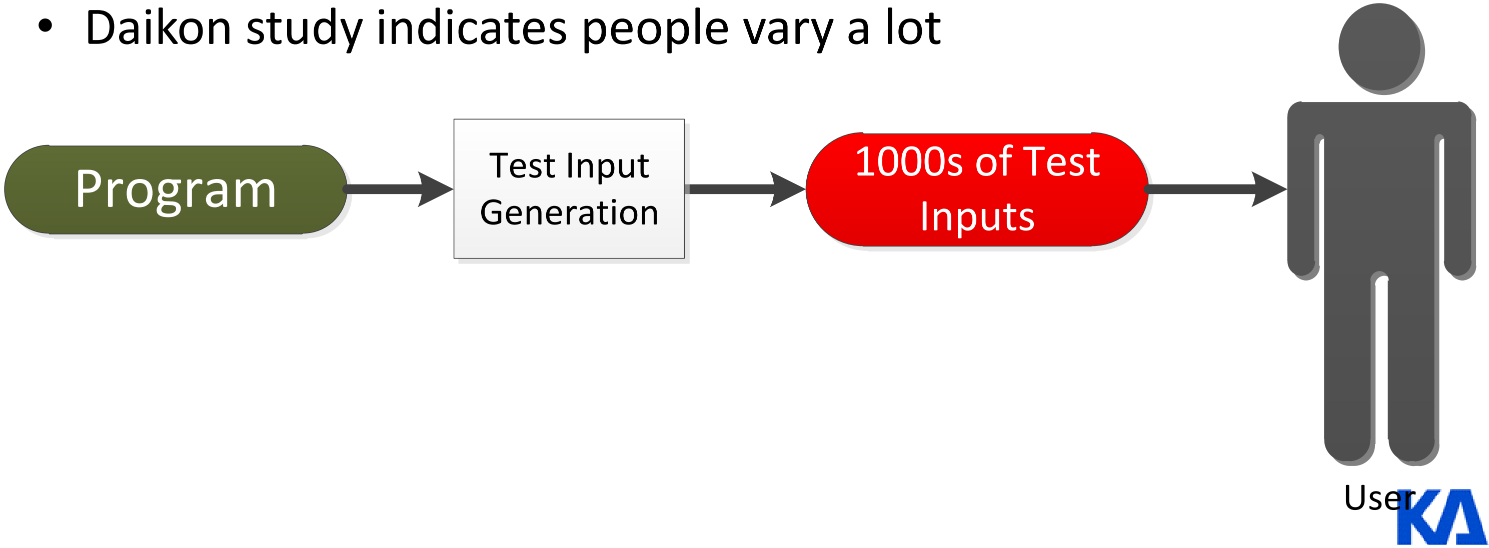
Tester-Centric Automated Testing

- Leads to ideas for future work
- Several problems/issues left
  - Method of supporting oracle selection is coarse at best
  - **Test input and oracle generation always separate**


Given test inputs, construct oracle
Oracle
Test Inputs
Given oracle construct test inputs

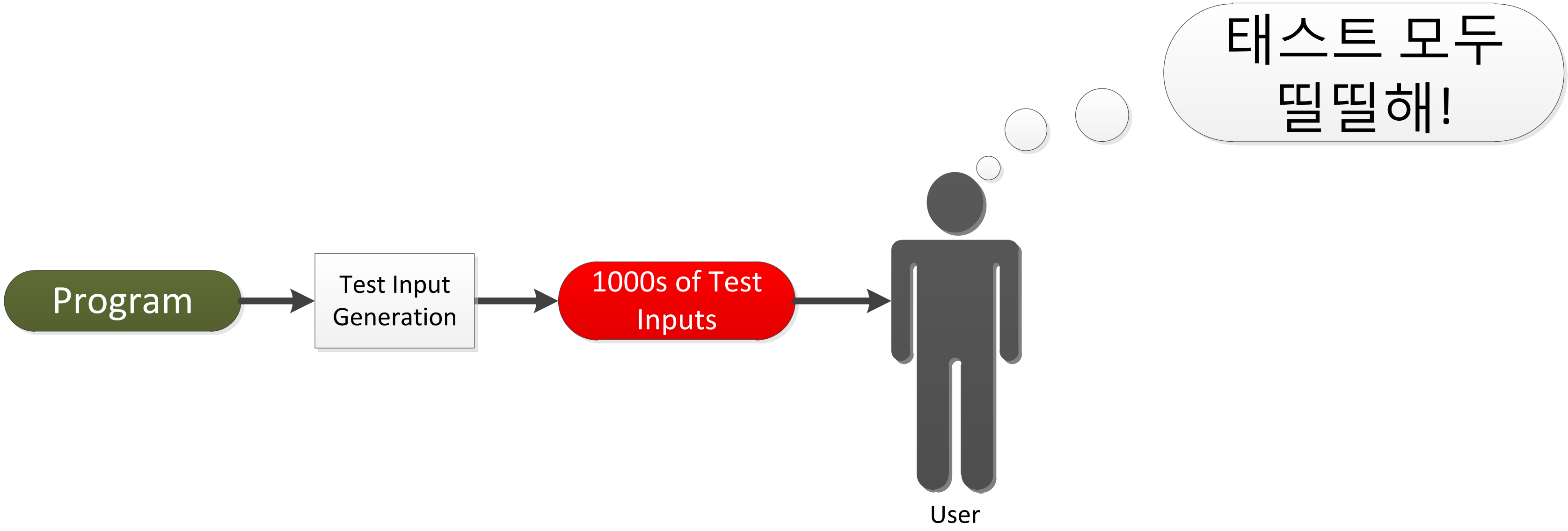
Tester-Centric Automated Testing

- Leads to ideas for future work
- Several problems/issues left
  - Method of supporting oracle selection is coarse at best
  - Test input and oracle generation always separate
  - **In generating inputs, no consideration of individual user preferences**
    - Lots of inputs, unclear user understands / wants them
    - Some work on simplfying inputs, but…
    - Daikon study indicates people vary a lot



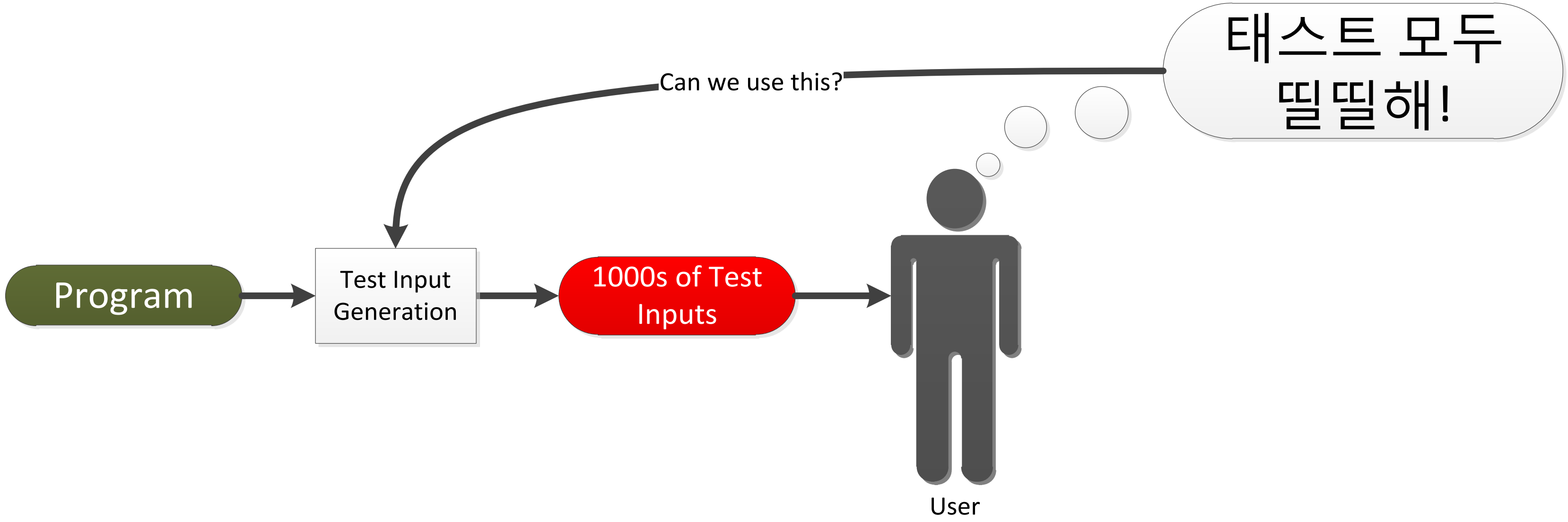
Tester-Centric Automated Testing

- Leads to ideas for future work
- Several problems/issues left
  - Method of supporting oracle selection is coarse at best
  - Test input and oracle generation always separate
  - **In generating inputs, no consideration of individual user preferences**



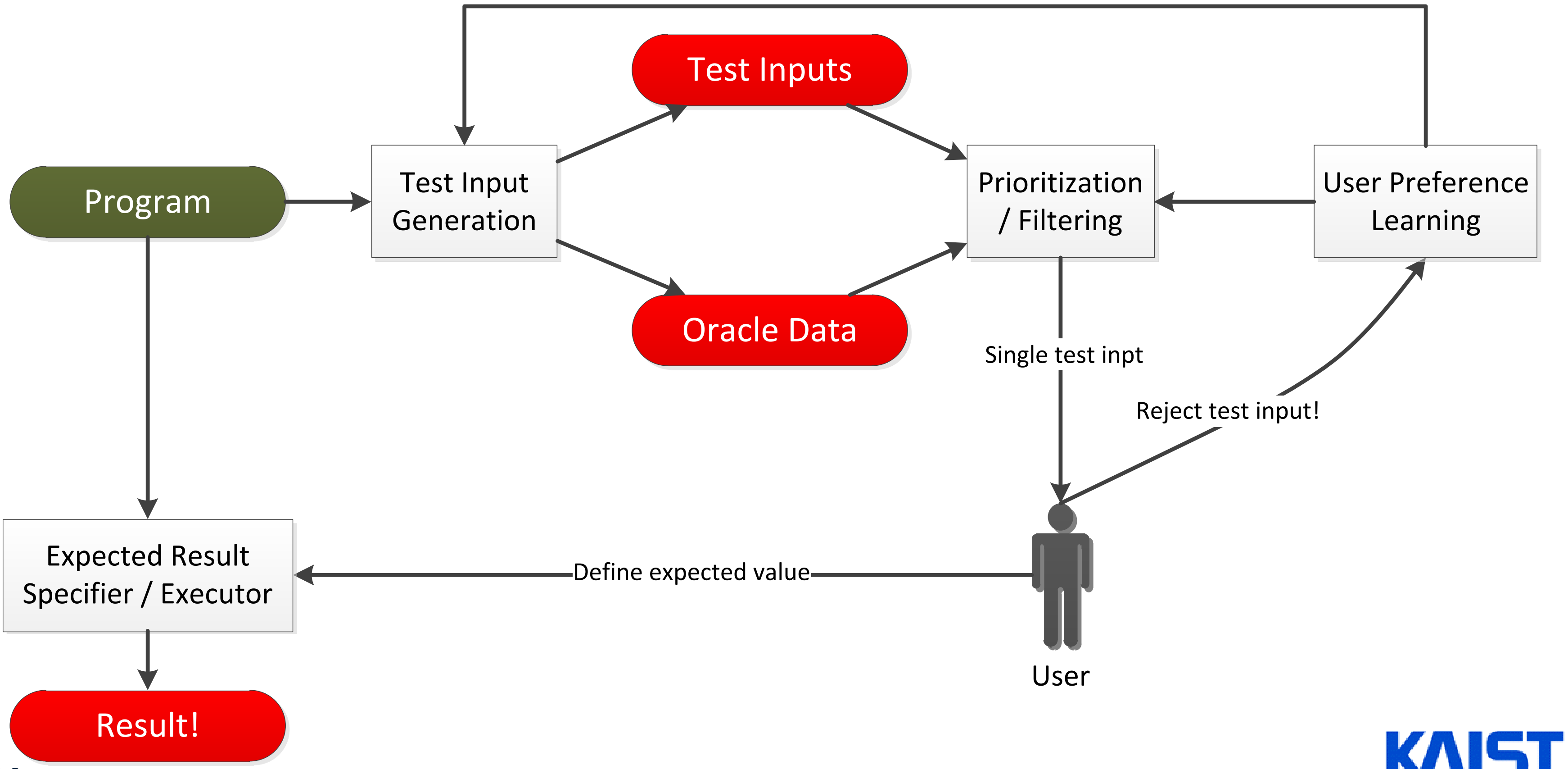
Tester-Centric Automated Testing

- Leads to ideas for future work
- Several problems/issues left
  - Method of supporting oracle selection is coarse at best
  - Test input and oracle generation always separate
  - **In generating inputs, no consideration of individual user preferences**

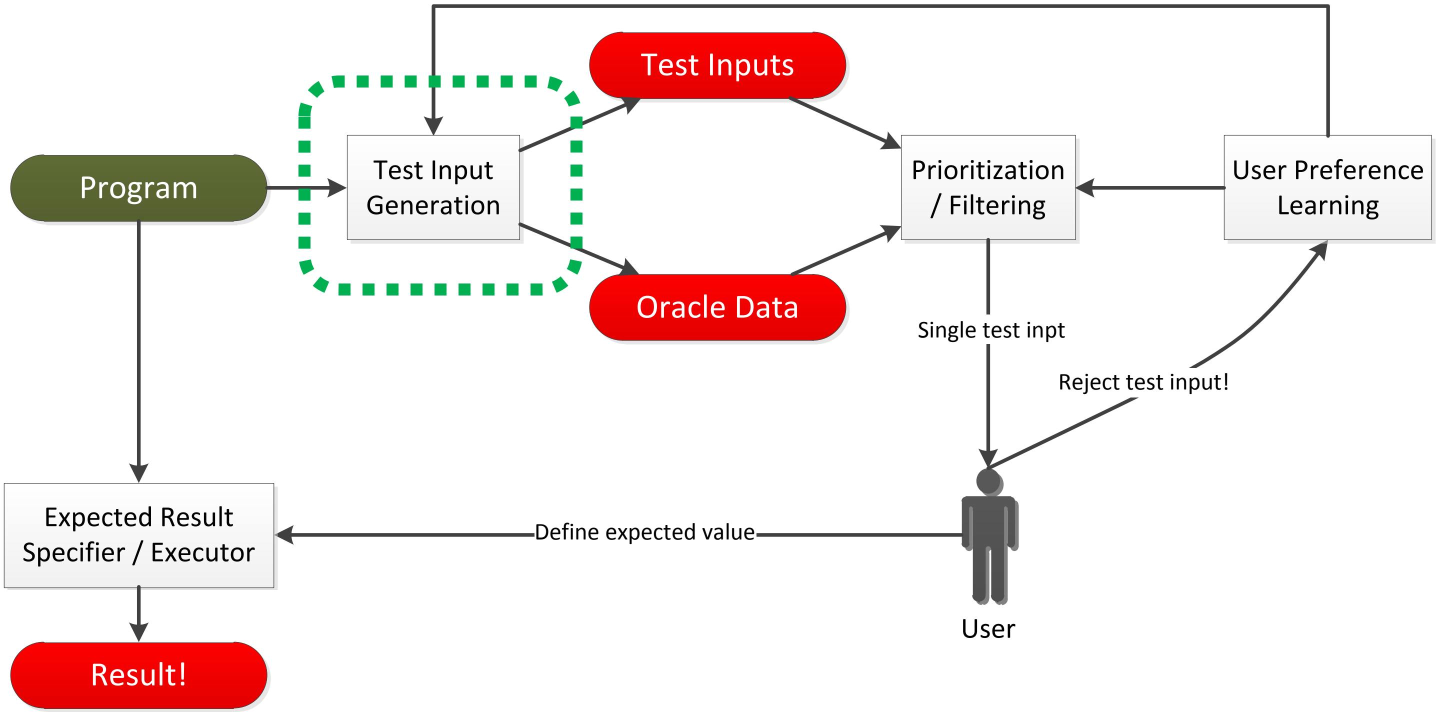


Tester-Centric Automated Testing

- Add together potential solutions, view of automated testing changes considerably
- More about optimizing for user preferences and saving user time

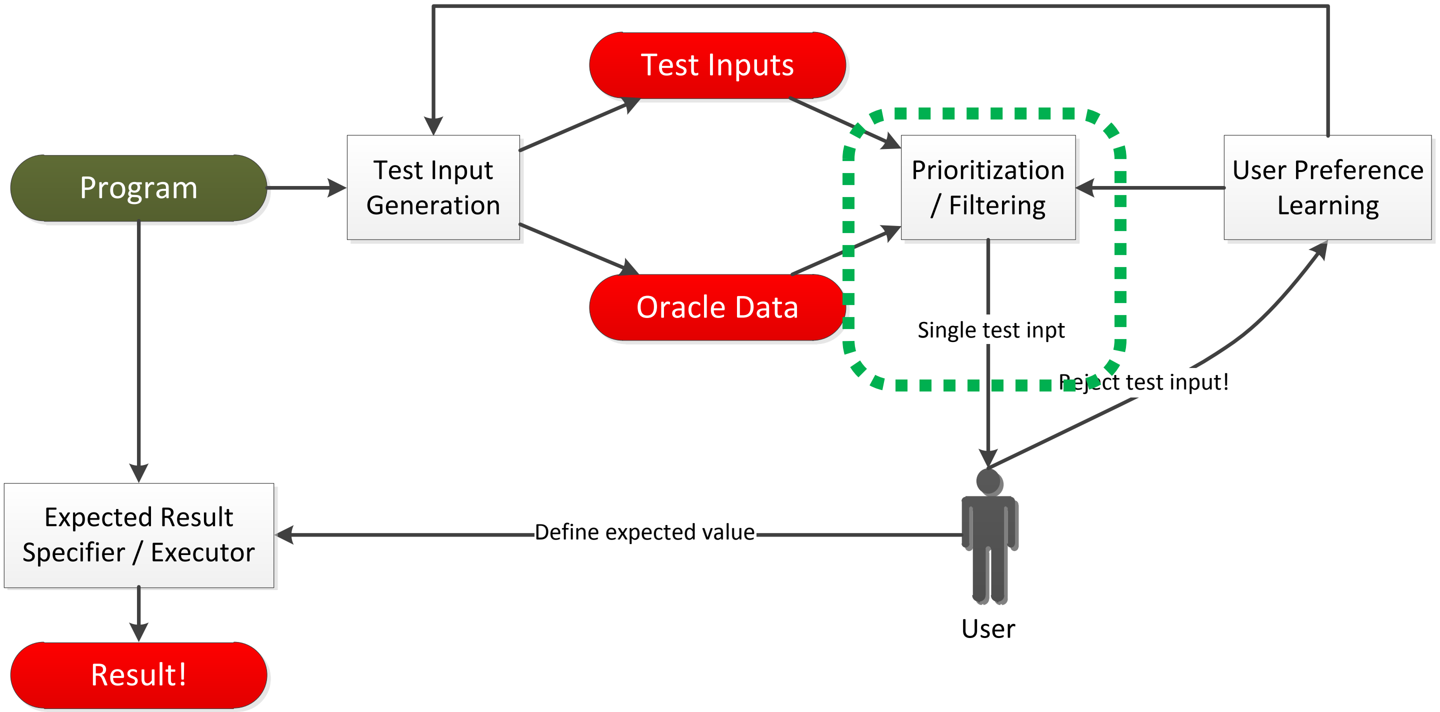


Tester-Centric Automated Testing



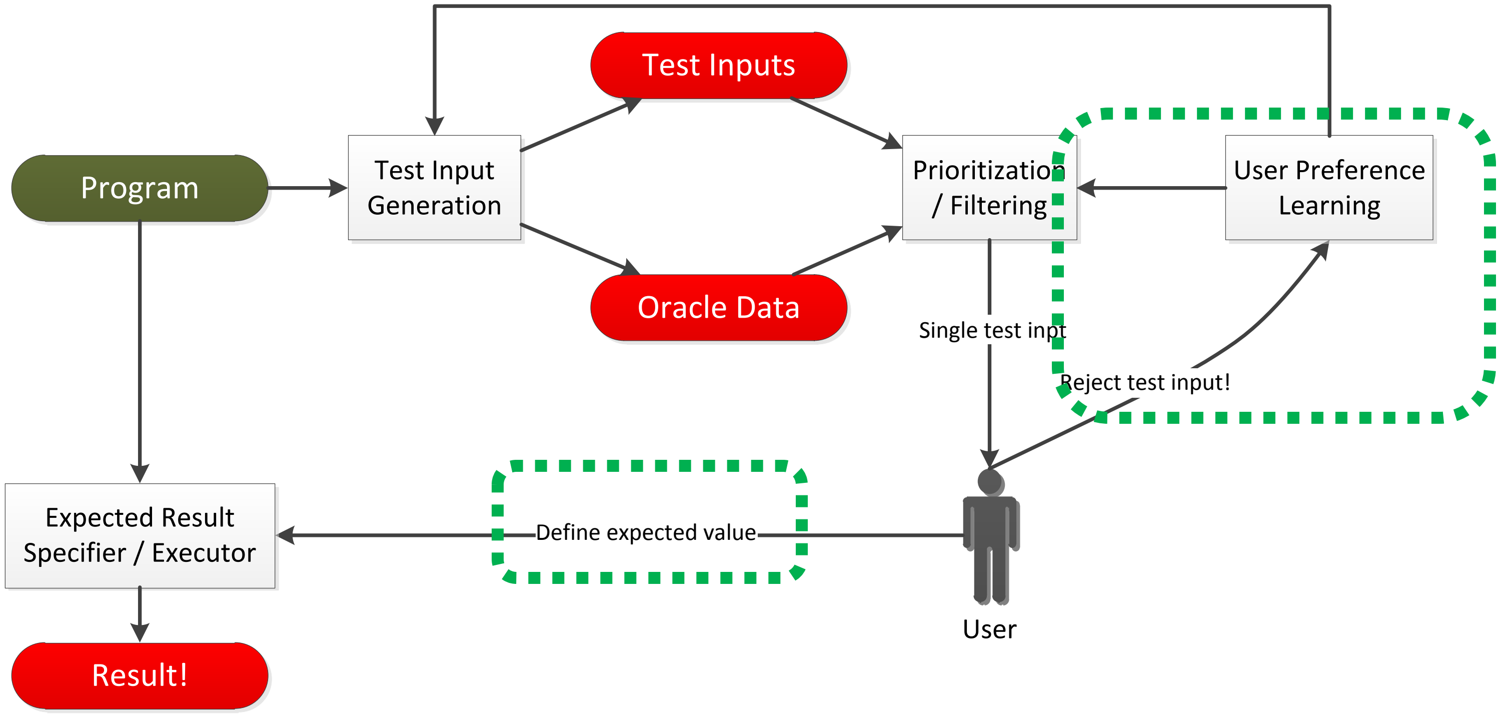
- Need fast, effective method of determining oracle data

Tester-Centric Automated Testing



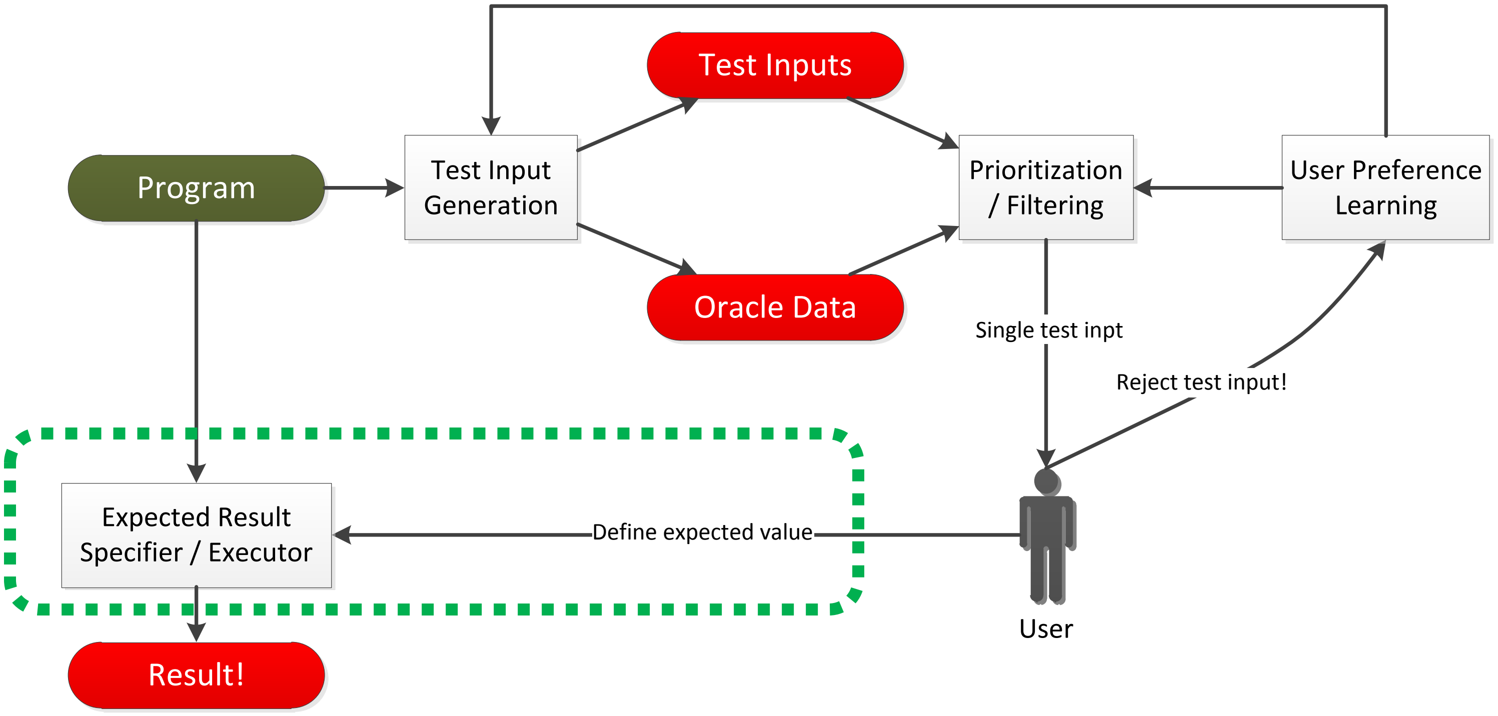
- Need fast, effective method of determining oracle data
- Prioritization / filtering is about maximizing result relative to cost of *user time*

Tester-Centric Automated Testing



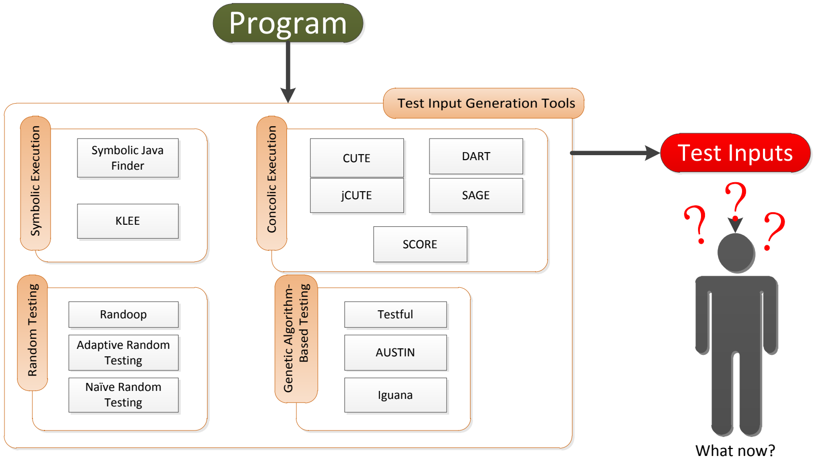
- Need fast, effective method of determining oracle data
- Prioritization / filtering is about maximizing result relative to cost of *user time*
- Must incorporate user preferences into generation / filtering process

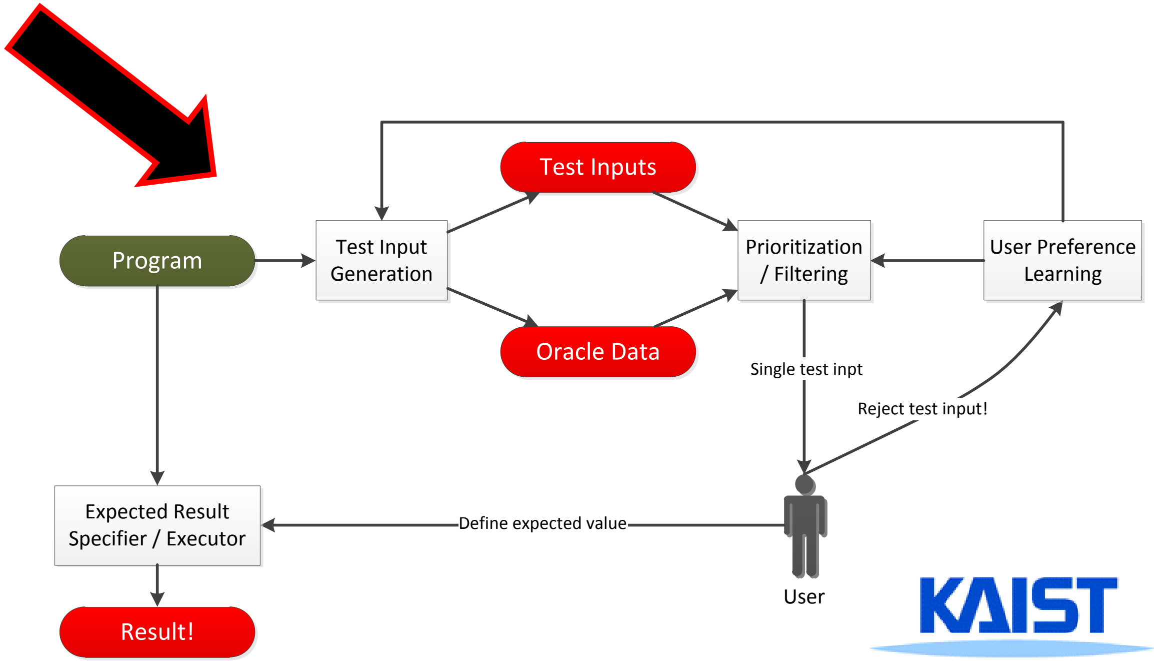
Tester-Centric Automated Testing



- Need fast, effective method of determining oracle data
- Prioritization / filtering is about maximizing result relative to cost of *user time*
- Must incorporate user preferences into generation / filtering process
- Need interface for users to work with

Now vs. Future





Questions

?