



PACIFIC NW
28TH ANNUAL
SOFTWARE
QUALITY
CONFERENCE

OCTOBER 18TH – 19TH, 2010



ACHIEVING
QUALITY
IN A COMPLEX
ENVIRONMENT

*Conference Paper Excerpt
from the*
CONFERENCE
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

Using Simple Automation to Test Complex Software

Harry Robinson, Microsoft

Software continues to grow more complex. Users want powerful features, they want those features to work smoothly, and they want those features delivered *yesterday*. Such demands make our industry a fascinating place to work, but they can make a test team's job a difficult challenge.

It might be natural to think that elaborate test systems will solve your problems; however those tools and infrastructures impose their own costs, and can distract you from your real mission of delivering great software.

Over the past year, Bing test teams have been experimenting with a *simpler* approach. Turning away from monolithic test infrastructures, we are finding that lightweight automation and heuristic oracles keep our tests flexible and productive while extending the reach of our exploratory testers.

Harry's keynote presentation will take the audience through the Bing team's journey to simplify and improve their testing — by offering the lessons learned, strange encounters during the process, and the encouraging results observed.

Harry Robinson is a Principal Software Design Engineer in Test (SDET) for Microsoft's Bing team. Harry has over twenty years of software development and testing experience at AT&T Bell Labs, Hewlett-Packard, Microsoft, and Google, as well as time spent in the startup trenches.

While at Bell Labs, Harry created a model-based testing system that won the AT&T Award for Outstanding Achievement in the Area of Quality. At Microsoft, he pioneered the model-based test generation technology, which won the Microsoft Best Practice Award. Harry holds two patents for software test automation methods, maintains the site www.model-based-testing.org, and speaks and writes frequently on software testing and automation issues.

using simple automation to test complex software

Harry Robinson
PNSQC 2010

Definitions

simple

- able to be done or understood quickly
- lacking decoration or embellishment

complex

- difficult to analyze, understand, or solve
- made up of many interrelated parts

Encarta® World English Dictionary

Software: Complexity and Quality

“Software entities are more complex for their size than perhaps any other human construct ...”

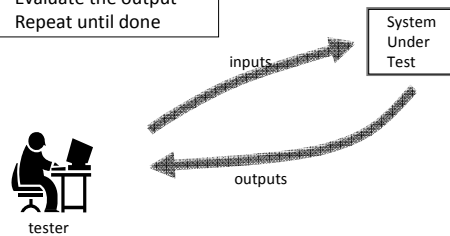
F. P. Brooks, *No Silver Bullet*

“Software complexity ... grows to the limits of our ability to manage that complexity.”

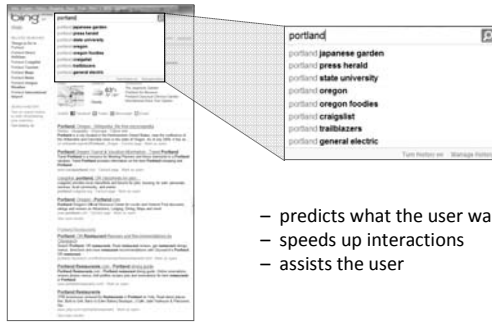
Boris Beizer, *Software Testing Techniques*

The Essence of Testing

1. Provide an input
2. Evaluate the output
3. Repeat until done



Welcome to AutoSuggest



- predicts what the user wants
- speeds up interactions
- assists the user

The Essence of Testing

1. Provide an input
2. Evaluate the output
3. Repeat until done

Input sources:

- static { d, de, def, stp, st p, portla, ...}
- random {qxr, mxyzptlk, lfjgo, sj3g, ...}
- exhaustive {aaaa, aaab, aaac, aaad, ...}
- patterns {<popular queries>, <DSATs>, ...}

The Essence of Testing

1. Provide an input
2. Evaluate the output
3. Repeat until done

Oracle sources:

- golden results
- eyeballs
- crashes
- heuristics

The Essence of Testing

1. Provide an input
2. Evaluate the output
3. Repeat until done

```
while (it's worth continuing)
{
    provide an input;
    evaluate the output;
    log the results;
}
```

Modes of Software Testing (1)

Manual Exploratory Testing



Pro

- Finds bugs
- Flexible

Con

- Labor-intensive
- Slow
- Limited range
- Biased by intuition

Modes of Software Testing (2)

Static Test Automation



Pro

- Fast
- Repeatable

Con

- Rigid
- Fragile
- Limited input
- Finds few bugs
- Heavyweight
- Distracting

Good AutoSuggestions Are ...

- Useful
 - Predictive
 - Fast
- Safe
 - No adult terms
 - No adult links
- Reasonable
 - No duplicates
 - No misspellings
 - No garbage text

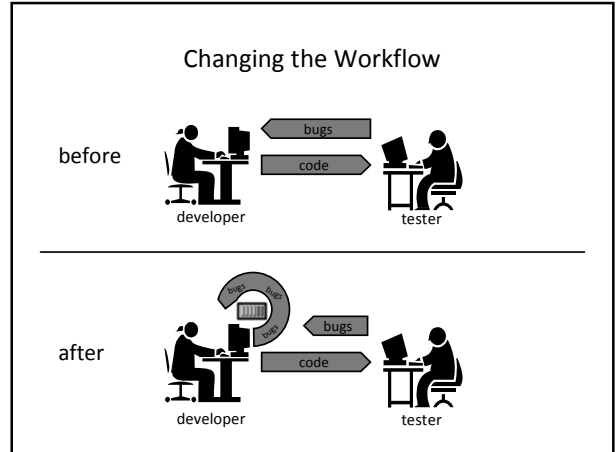
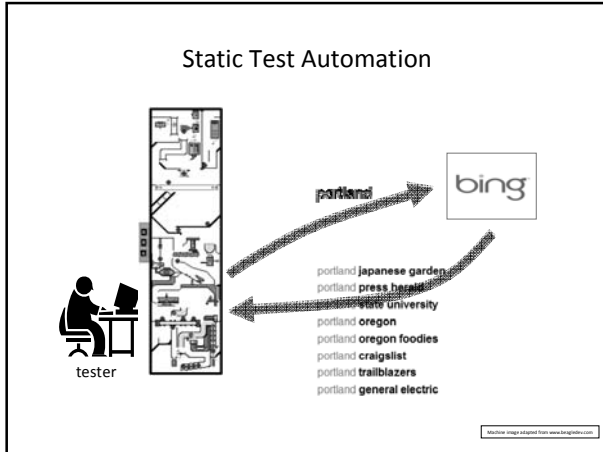
port
portland japanese garden
portland press herald

dagna
dagnabbit
dagna barrera

aaaaa
aaaaa
aaaaaaa aaa.aaa

Functional Tests

#	Input	Expected output
1	d	dictionary, disney channel, delta airlines, dell, ...
2	de	delta airlines, dell, detroit lions, derek jeter, ...
3	def	definitions, defender, def jam, defraggler, ...
4	stp	st pete times, stp, st paul pioneer press, ...
5	st p	st pete times, st paul pioneer press, st patricks day, ...
6	portla	portland japanese garden, portland press herald, ...



Heuristic Test Oracles

- Verify some individual results
- Check other results using
 - simple algorithms
 - consistency checks

Douglas Hoffman, *An Updated Taxonomy for Test Oracles*

Static oracle
 $\text{sqrt}(17) = 4.1231056$

Heuristic oracle
 $\text{sqrt}(N) * \text{sqrt}(N) = N$

Testing AutoSuggestion with Heuristics

Problem: How can we verify an enormous input space?

Input	Suggestion	Reason
def	def jam	prefix
nestb	bestbuy.com	spell correction
dond	nbc.com	?

Heuristic: Autosuggestions should be "explainable"

Results:

- Covered 22 million triggers in 7 days
- Detected 2000 bad suggestions

Emergent Test Opportunities

- Dead suggestions
- Suggestions with empty results
- Suggestions with adult results
- Misspellings
- Meaningless suggestions
- Powerful unit testing

New Risks

- Poor input generation:
 - insufficient coverage
- Poor oracling:
 - false positives
 - false negatives

Lessons Learned

- Simplicity can produce a virtuous cycle
- Oracles can be non-binary
- Humans and computers can coexist
- What seems simple may not be
- Simple != Easy
- Testing == Engineering

Acknowledgments

I am happy to acknowledge the work of Todd Paul, Reena Agarwal, Edward Unpingco, DJ Ramakrishnappa, Dragos Boia, Asraful Islam, Cyril Bouanna, Jing Tan, Andrey Yegorov, Nick Karnik and many other Bing team members in supporting, prototyping, experimenting with and developing the methods described in this presentation.