Obstacles and Opportunities for Model-Based Testing in an Industrial Software Environment

Harry Robinson, Test Architect
Microsoft Enterprise Management Division
harryr@microsoft.com
Of Mousetraps and Bugtraps

Build a better mousetrap
and the world will beat a path to your door.
- Ralph Waldo Emerson

... assuming the world understands good mousetraps...
- Harry Robinson
Typical Industry Lifecycle

Spec → Dev → Test

bugs

bugs
Typical Industry Tester

- Bachelor's: 54%
- Master's: 17%
- Some college: 28%
- Doctorate: 1%

Source: STQE Salary Survey 2002
Introducing MBT into the Industry

- Relative Advantage
- Compatibility
- Complexity
- Trialability
- Observability

(from Diffusion of Innovations, by Everett Rogers)
“the greater the relative advantage of an innovation, the more rapid the rate of its adoption will be.”
Is model-based testing better than existing methods?

Target this group
Ways of demonstrating relative advantage (in order of increasing credibility)

- Demos
- External case studies
- Internal case studies
- Working code
Compatibility

“an idea that is not compatible with the prevalent values and norms of a social system will not be adopted as rapidly as an innovation that is compatible.”
Does model-based testing integrate with existing methods?

"...the primary function of the test group is to find bugs..." - Cem Kaner
Does model-based testing integrate with existing methods?

Management expects a bug graph to look like this

But your bug graph looks like this

“The higher you go in management, the more you rely on numbers.” - Richard Bender
Compatibility

- Target the most technical people
- Target the least invested people
- Start very, very small:
  - nano-pilot
  - micro-pilot
  - mini-pilot
  - pilot
“new ideas that are simpler to understand will be adopted more rapidly than innovations that require the adopter to develop new skills.”
Is model-based testing difficult to understand?

\[
\text{project} : \text{OZSpec} \rightarrow \text{UMLDiagram}
\]

\[
\forall (oz, uml) : \text{project} \cdot
\begin{align*}
\{ c : oz \cap \text{Classdef} \land c.\text{name} \} &= \{ c : uml.\text{classes} \land c.\text{name} \} \cup \forall c_1, c_2 : oz \cap \text{Classdef} \land \exists c' : uml.\text{classes} \land c'.\text{name} = c_1.\text{name} \\
&\land c'.\text{attris} = \{ \text{cls} : \text{Classdef} \mid \text{cls} \in oz \land \text{cls}.\text{name} \in c_1.\text{name} \} \\
&\land \forall c_1.\text{state}.\text{depart} \\
&\land o.\text{ops} = \{ o : \text{OOpDef} \mid o \in c_1.\text{ops} \land o.\text{name} \} \\
&\land c_2.\text{name} \in \{ t : \text{ran} c_1.\text{state}.\text{depart} \land t.\text{name} \} \Rightarrow \\
&\exists (c'_1, c'_2) : \text{uml.}\text{agg} \land c'_1.\text{name} = c_1.\text{name} \\
&\land c'_2.\text{name} = c_2.\text{name} \\
&\land c'_2.\text{name} \in \{ \text{inh} : \text{dom} c_1.\text{inherit} \land \text{inh}.\text{name} \} \Rightarrow \\
&\exists (c'_1', c'_2') : \text{uml.}\text{inh} \land c'_1'.\text{name} = c_1.\text{name} \\
&\land c'_2'.\text{name} = c_2.\text{name}
\end{align*}
\]

And remember this diagram!
- Complexity

- Courses/workshops
- Working examples
- Visualization/tools
“new ideas that can be tried on the installment plan will generally be adopted more quickly than innovations that are not divisible.”
Is it easy for people to experiment with model-based testing?

- Education
- Time
Is it easy for people to experiment with model-based testing?

- Working models
- Tool support
- “Extreme modeling”
“the easier it is for individuals to see the results of an innovation, the more likely they are to adopt.”
Are the benefits of model-based testing easily visible?

- How can you count bugs you prevented?
- Should you count how many test cases you’ve generated?
- Emphasize end product, not in-process metrics
- Replace bad metrics with better metrics
  - Bad: bug counts, test case counts
  - Better: code coverage, spec coverage
- Sway technical advisors
Surprises

- Process Improvement Groups
- Test Tool Groups
- Recruiters
- Skill Set Challenges
- Developers
- Researchers
- Volunteers
- Interns
Where we’ve come from

Model-based testing email alias
Where we are now

Microsoft Best Practice Award in Software Testing 2001
What appears to be in our future

- More powerful modeling tools (e.g. AsmL)
- Increased integration into test infrastructure
- Better test generation tools
- Advanced test training
- Changes in recruiting
Some closing words …

As ease of use becomes increasingly important to customers, the complexity of generating complete test cases is also exploding and is a key challenge for test teams.

A systematic and automatic way of defining the test matrix is needed everywhere and that is exactly what model-based testing provides.

**Model-based testing** will be the primary method of creating test plans and test cases.

Craig Zhou  
Director, Microsoft Windows Test Infrastructure
www.model-based-testing.org

thank you