Automation in 20 Minutes

Michael Bolton
DevelopSense
http://www.developsense.com

TESTREK Toronto Managers' Workshop
October, 2009

Testing Isn't Just *Checking*

- Checking is a process of confirming and verifying existing beliefs
  - Checking can (and I argue, largely should) be done mechanically
  - It is a *non-sapient* process

See http://www.developsense.com/2009/08/testing-vs-checking.html

What *IS* Checking?

- A *check* has three attributes
  - It requires an *observation*
  - The observation is linked to a *decision rule*
  - The observation and the rule can be applied

without sapience

Oh no! What Is *Sapience*?

- A sapient activity is one that requires a thinking human to perform
- A non-sapient activity can be performed by
  - a machine (quickly and precisely)
  - or by a human that has decided NOT to think (slowly and fallibly)
  - looks like machines win there, right?
- BUT our job is not merely to test for repeatability, but also for *adaptability and value*

Automation Can't…

recognize new risks investigate speculate
empathize anticipate predict suggest
recognize refocus judge project
appreciate contextualize elaborate
become resigned strategize evaluate
teach question charter assess
learn get frustrated
reframe work around a problem
invent
model make conscious decisions
troubleshoot collaborate refine resource

Automation Can't…

recognize new risks investigate speculate
empathize anticipate predict suggest
recognize refocus judge project
appreciate contextualize elaborate
become resigned strategize evaluate
teach question charter assess
learn get frustrated
reframe work around a problem
invent
model make conscious decisions
troubleshoot collaborate refine resource

FEEL

Automation Can't…

recognize new risks investigate speculate

empathize anticipate predict suggest

recognize focus judge project

appreciate contextualize elaborate

become resigned strategize evaluate

question chart assess

teach learn get frustrated

reframe work around a problem

invent

make conscious decisions

model

troubleshoot collaborate refine resource

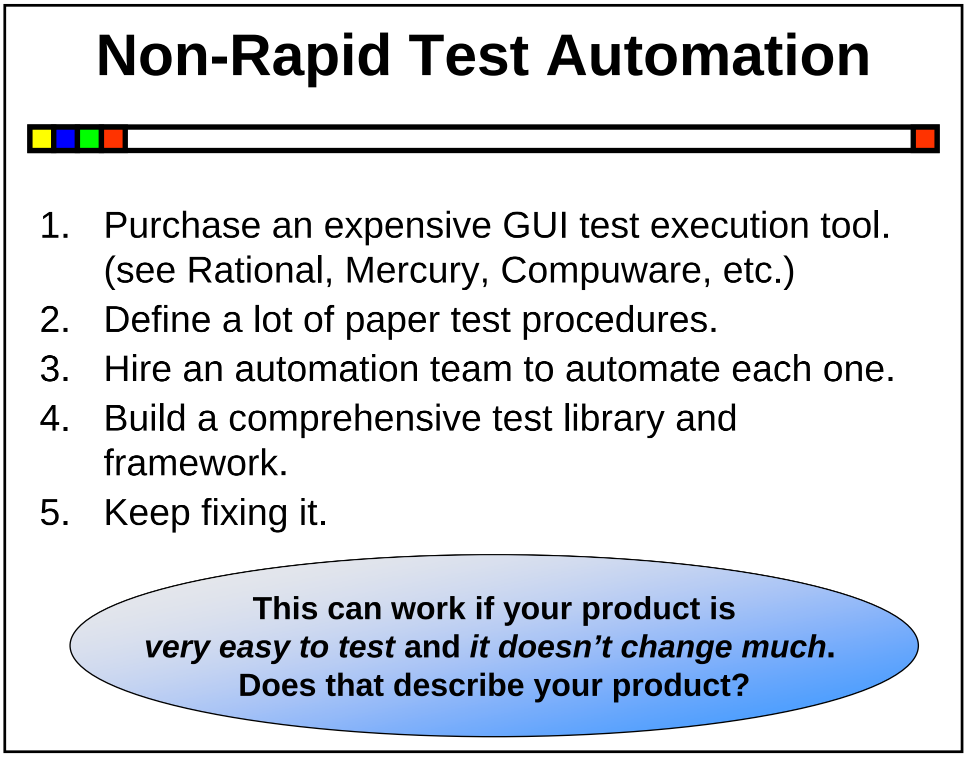
FEEL

THINK

Testing IS Exploring

- Testing as I see it is all about exploration, discovery, investigation, and learning
  - Testing can be assisted by machines, but can't be done by machines alone
  - It is a *sapient* process

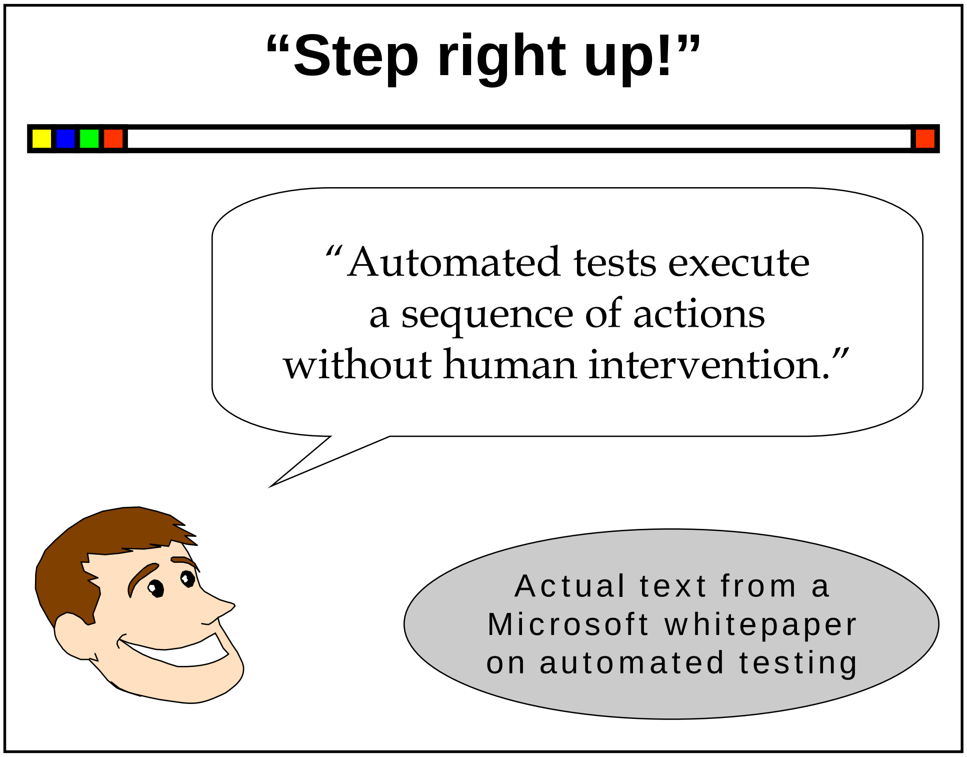
See http://www.developsense.com/2009/08/testing-vs-checking.html



Non-Rapid Test Automation

1. Purchase an expensive GUI test execution tool. (see Rational, Mercury, Compuware, etc.)
2. Define a lot of paper test procedures.
3. Hire an automation team to automate each one.
4. Build a comprehensive test library and framework.
5. Keep fixing it.

This can work if your product is
***very easy to test* and *it doesn't change much*.**
Does that describe your product?



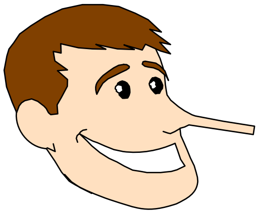
"Step right up!"

"Automated tests execute
a sequence of actions
without human intervention."

Actual text from a
Microsoft whitepaper
on automated testing

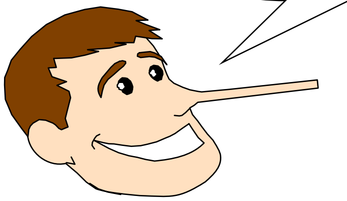
“Step right up!”

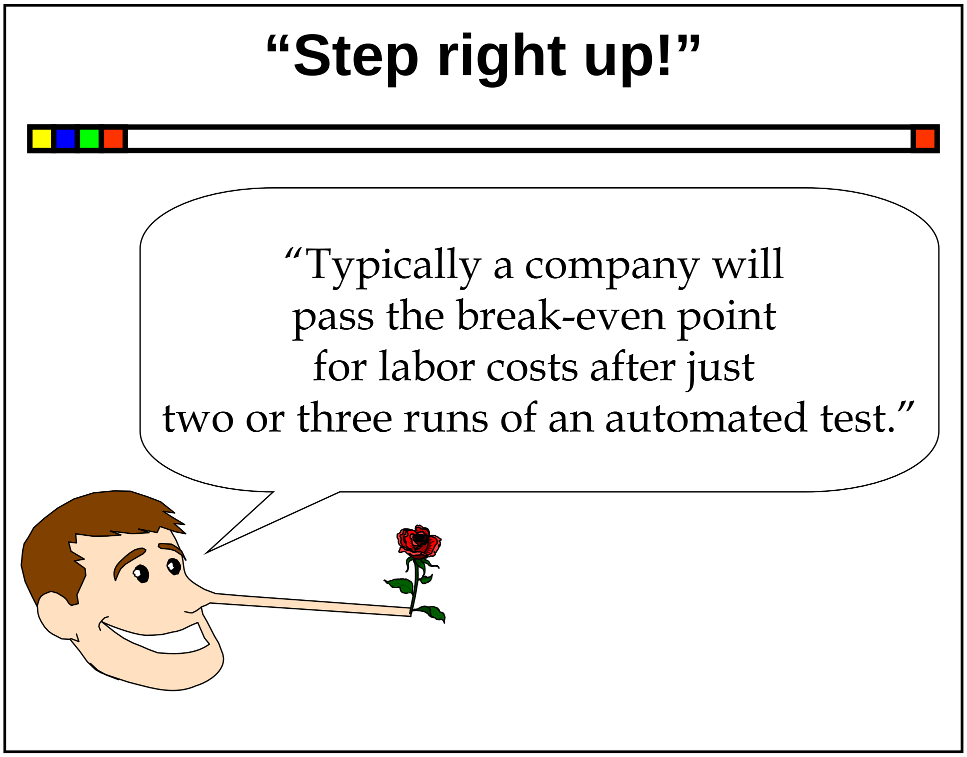
“This approach helps eliminate human error, and provides faster results. ”



“Step right up!”

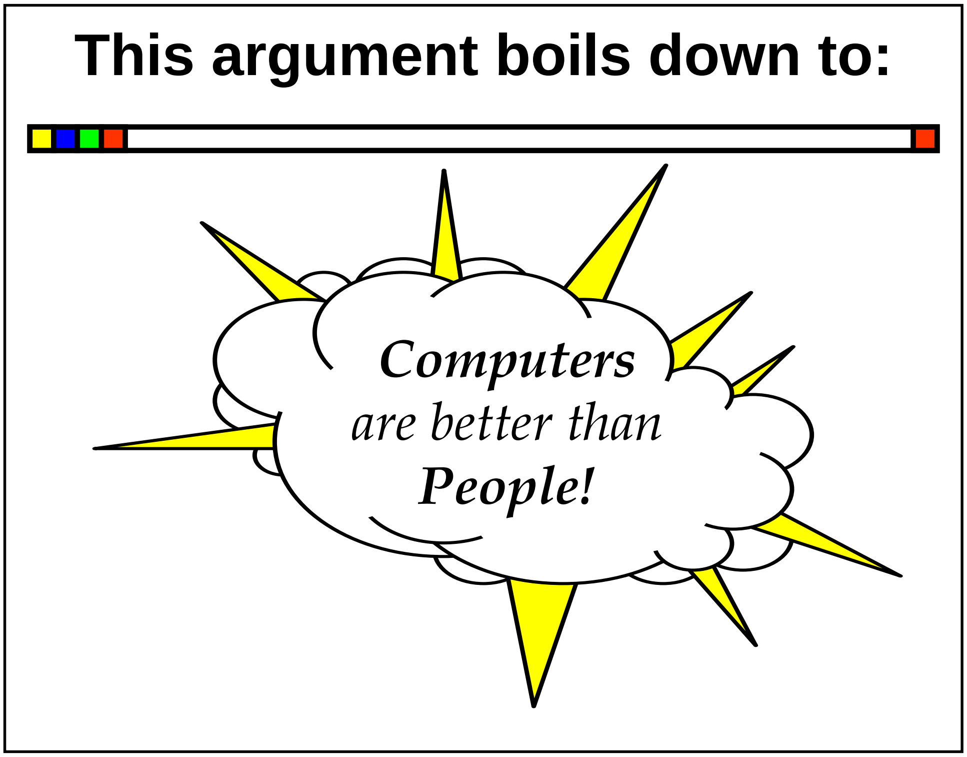
“Since most products require tests to be run many times, automated testing generally leads to significant labor cost savings over time. ”





“Step right up!”

“Typically a company will
pass the break-even point
for labor costs after just
two or three runs of an automated test.”



This argument boils down to:

Computers
are better than
People!

Reckless Assumptions

1

Testing is a “sequence of actions”

Actually, testing is better viewed as an interactive cognitive process.

Reckless Assumptions

2

Testing means repeating the same actions over and over.

Testing has repetitive elements,
but it isn’t as repetitive as it seems,
and variety is critical to the process.

Reckless Assumptions

3

We can automate testing actions.

Actually, in most cases we can only automate a subset of testing actions, for the rest, we need human testers.

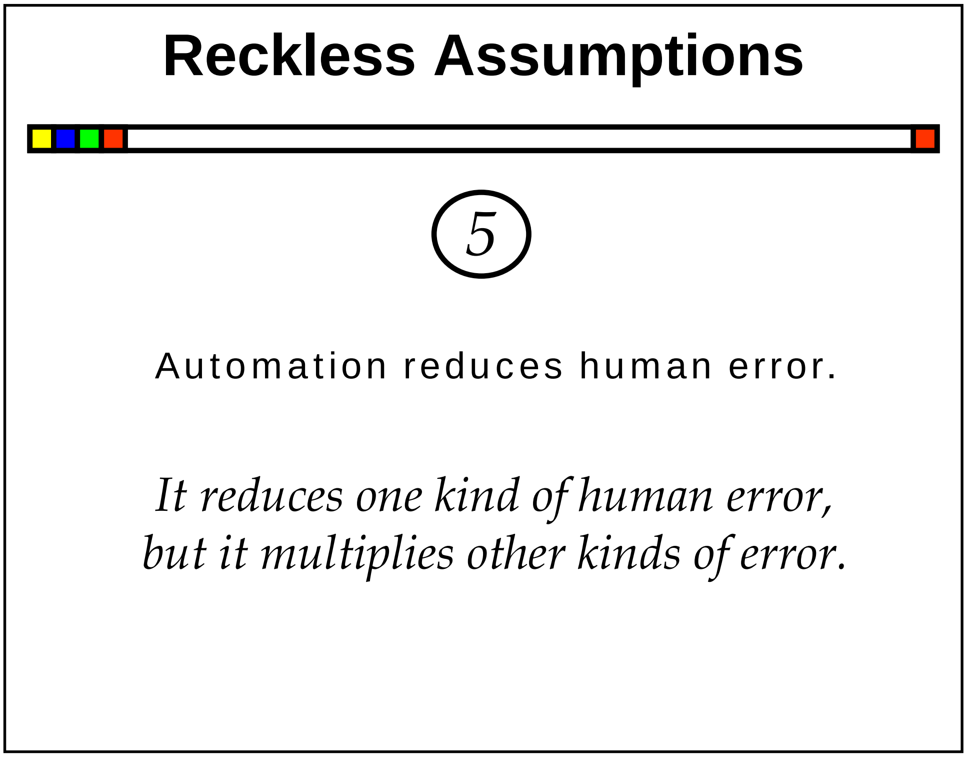
Reckless Assumptions

4

An automated test is faster,
because it needs no human intervention.

Faster simulation of keystrokes & clicks is not the same as faster testing.

Human intervention is reduced during execution, but increased during diagnosis.

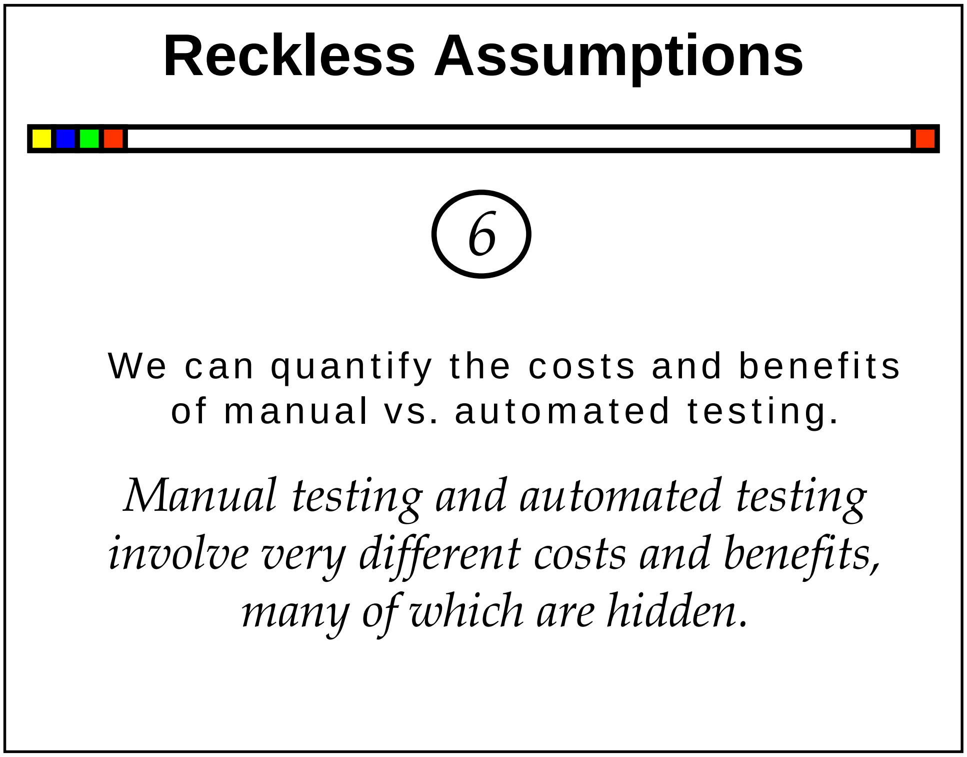


Reckless Assumptions

5

Automation reduces human error.

It reduces one kind of human error, but it multiplies other kinds of error.

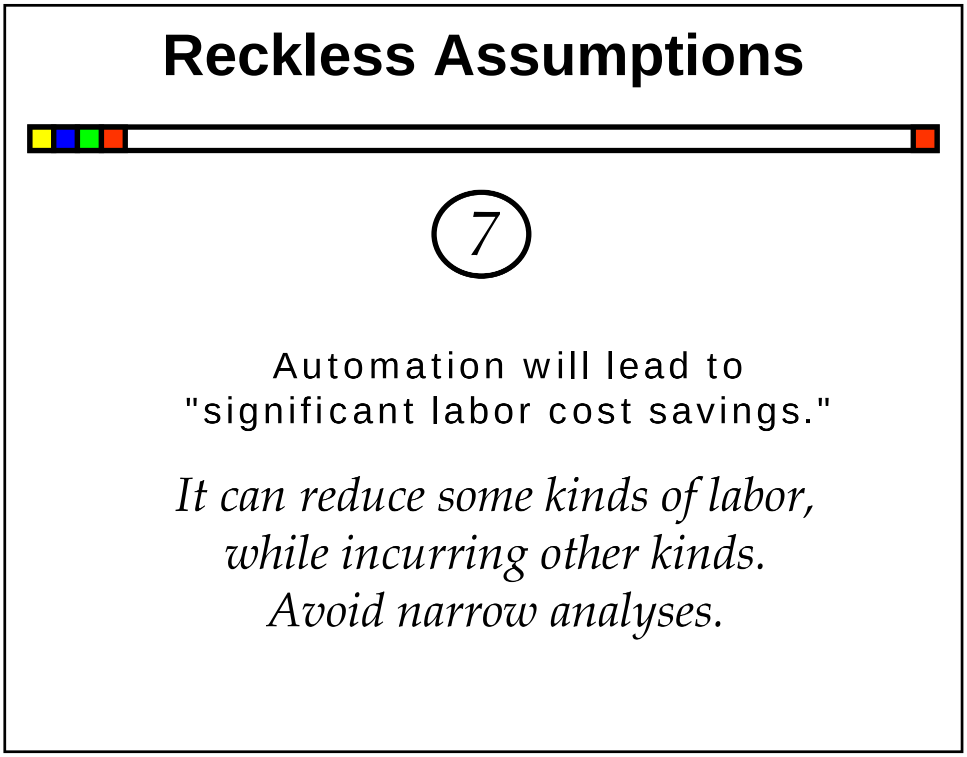


Reckless Assumptions

6

We can quantify the costs and benefits of manual vs. automated testing.

Manual testing and automated testing involve very different costs and benefits, many of which are hidden.

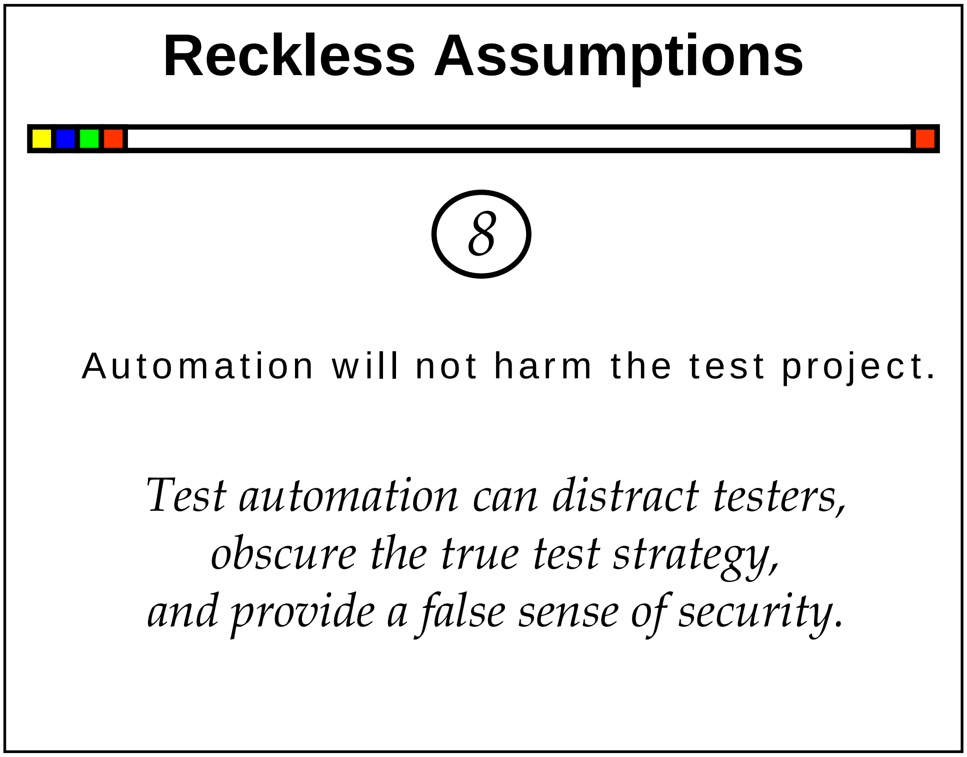


Reckless Assumptions

7

Automation will lead to
"significant labor cost savings."

It can reduce some kinds of labor,
while incurring other kinds.
Avoid narrow analyses.



Reckless Assumptions

8

Automation will not harm the test project.

Test automation can distract testers,
obscure the true test strategy,
and provide a false sense of security.

Towards a better model of test automation…

- Test automation is…

Any use of tools to support testing.

Tool-Supported Testing

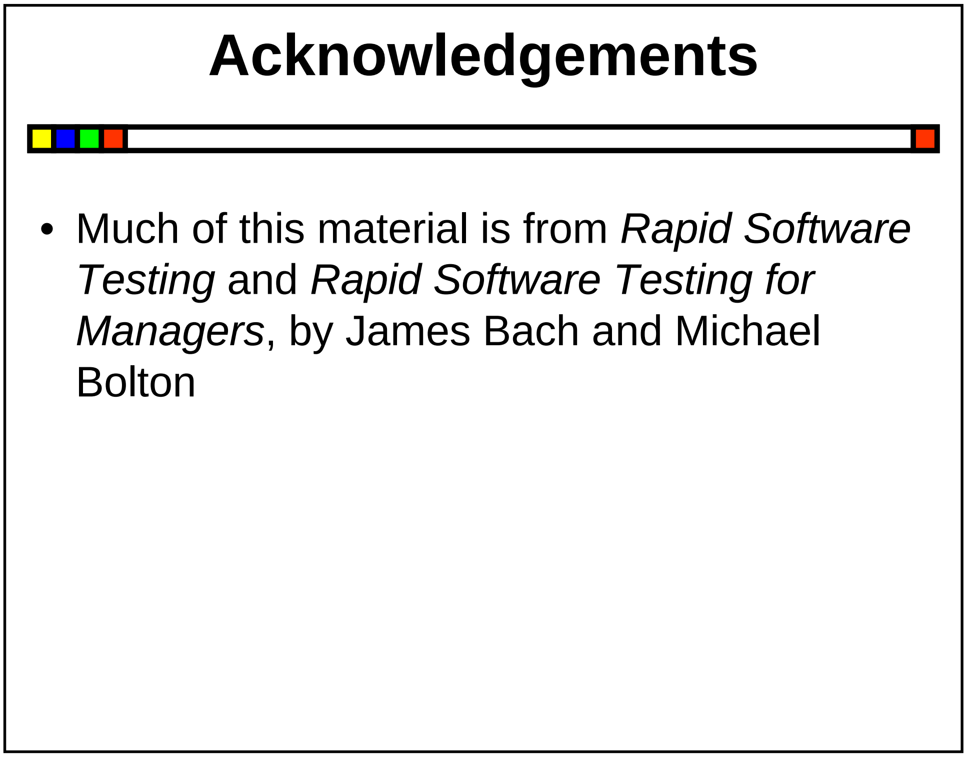
- **Test generation (data and script generators).** Tools might create specialized data such as randomized email messages, or populate databases, or generate combinations of parameters that we'd like to cover with our tests.
- **System configuration.** Tools might preserve or reproduce system parameters, set systems to a particular state, or create or restore "ghosted" disk drives.
- **Simulators.** Tools might simulate sub-systems or environmental conditions that are not available (or not yet available) for testing, or are too expensive to provide live on demand.
- **Test execution (harnesses and test scripts).** Tools might operate the software itself, either simulating a user working through the GUI, or bypassing the GUI and using an alternative testable interface.
- **Probes.** Tools might make visible what would otherwise be invisible to humans. They might statically analyze a product, parse a log file, or monitor system parameters.
- **Oracles.** An oracle is any mechanism by which we detect failure or success. Tools might automatically detect certain kinds of error conditions in a product.
- **Activity recording & coverage analysis.** Tools might watch testing as it happens and retrospectively report what was and was not tested. They might record actions for later replay in other tests.
- **Test management.** Tools might record test results; organize test ideas or metrics.

Test tools are all over the place.

- **MSDN Universal Library**
- **Any Microsoft development tool** (they always include useful utilities)
- **Microsoft compatibility toolkit** and other free tools (www.microsoft.com)
- **Web-based web testing resources** (HTML checkers, accessibility analyzers)
- **Windows Resource Kits** (available from Microsoft)
- **Scripting languages** (e.g. Perl, Ruby, TCL) and associated libraries
- **Shareware repositories** (www.download.com)
- **O/S monitoring tools** (www.sysinternals.com)
- **Open source testware** (www.opensourcetesting.org, www.sourceforge.com)
- **Spyware for monitoring exploratory tests** (www.spectorsoft.com)
- **The cubicle next door…** (someone else in your company has a tool for you)

Key Points:

- Testing is intellectual, not just clerical.
- Test automation is software development.
- Automation skill isn't cheap.
- Automation projects aren't cheap.
- Automation tends to worsen a poor test process.
- Test automation is a promising idea that often falls far short of its promise.
- *I mean to be cautionary, not discouraging: You can be successful in test automation if you successfully cope with these problems.*



Acknowledgements

- Much of this material is from *Rapid Software Testing* and *Rapid Software Testing for Managers*, by James Bach and Michael Bolton