

Spreadsheets Need Testing Too. Finding Billion Dollar Bugs

STARCanada

Gregory Pope, CSQE

April 8, 2014

Toronto, Ontario, Canada

 Lawrence Livermore
National Laboratory

LLNL-PRES-648155

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC



Eight Expensive Spreadsheet Mistakes

- Fidelity Minus Sign Mistake 1/1995\$2.6 Billion
- TransAlta “Clerical Error” 6/2003\$24 Million
- Fannie Mae “Honest” Mistake 10/2003\$1.3 Billion
- University of Toledo Projected Revenue 5/2004\$2.4 Million
- Red Envelope 3/2005Shares plunge 28%
- Think and Do 3/2005number of bachelor degrees 11% instead of 20% in Virginia
- Kodak Restates 11/2005 severance pay error\$11 Million
- Westpac Profit 11/2005.....Trading halted for early release of profits

Eight of the Worst Spreadsheet Blunders, Thomas Wailgum, August 17, 2007
http://www.cio.com/article/131500/Eight_of_the_Worst_Spreadsheet_Blunders?page=1&taxonomyId=3000



Minus Sign Mistake - \$2.6 B

- Copy from master spreadsheet to one for accountant
- Accountant omitted a minus sign from a \$1.3B capital loss, so it was counted as a capital gain.
- The net error was the dividend estimate was \$2.6B too high
- Lesson Learned – Differentiate gain and losses, have independent check



Clerical Error \$24M

- Canadian company bought more power than needed from US company at a higher price.
- Cut and paste error not caught on spreadsheet during sort and ranking of bids.
- Lesson learned: Have another employee double check documentation



FannieMae[®]

Honest Mistake \$1.3B

- Mistakes made in implementation of new accounting standard.
- Lesson learned: Have a financial peer review the documentation.



Projected Revenue - \$2.4M

- Typo in formula overstated the funds available for use.
- Lesson learned: Extra scrutiny and review. User training.

28% Share Value Drop

- Overestimation of gross margins led to forth quarter over projection of profits.
- Number misrecorded in cell.
- Lesson learned: More quality control

Number degrees 11% instead of 20%

- Researchers at Virginia Tech cut and paste error causing the number of population over 25 with bachelors degrees in a region to be lower than actual.
- Lesson learned: Have another employee check the work



Restates Severance - \$11M

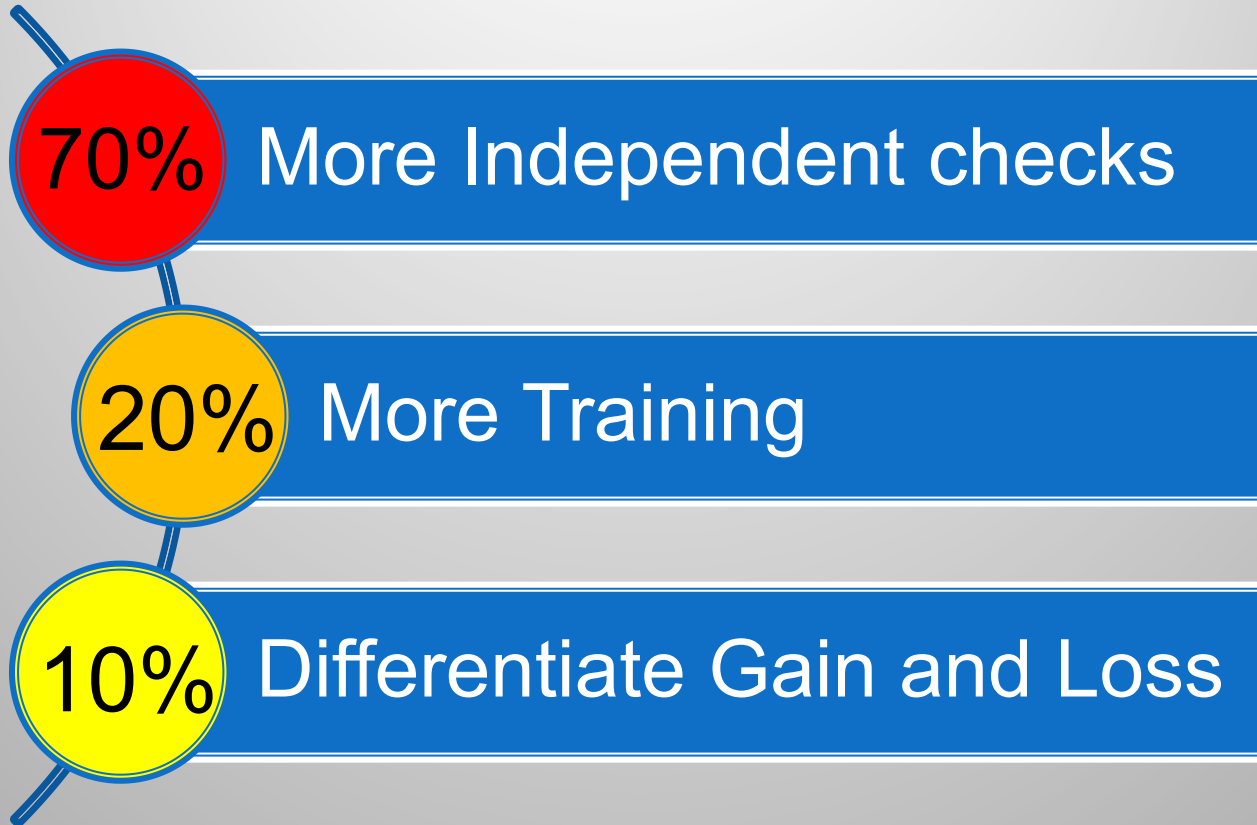
- Over stated severance pay due. Error was on one employee's severance pay, there were too many zeros.
- Lesson learned: Lack of data quality control



Westpac Bank Profit Trading Suspended

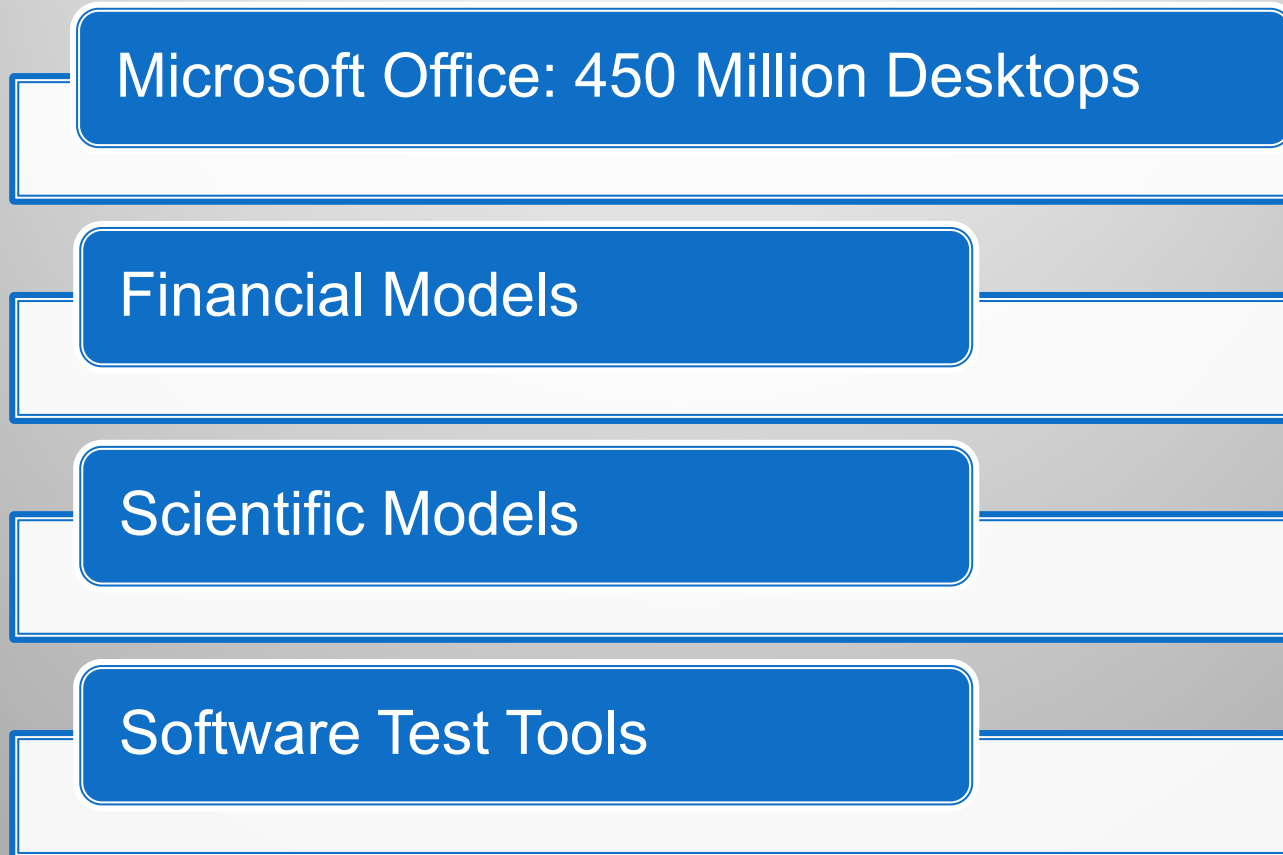
- Released the results of next quarter profits early by putting them into an existing spreadsheet and then hiding them by making the cell fill black.
- Lesson learned: Additional training.

Lessons Learned Summary



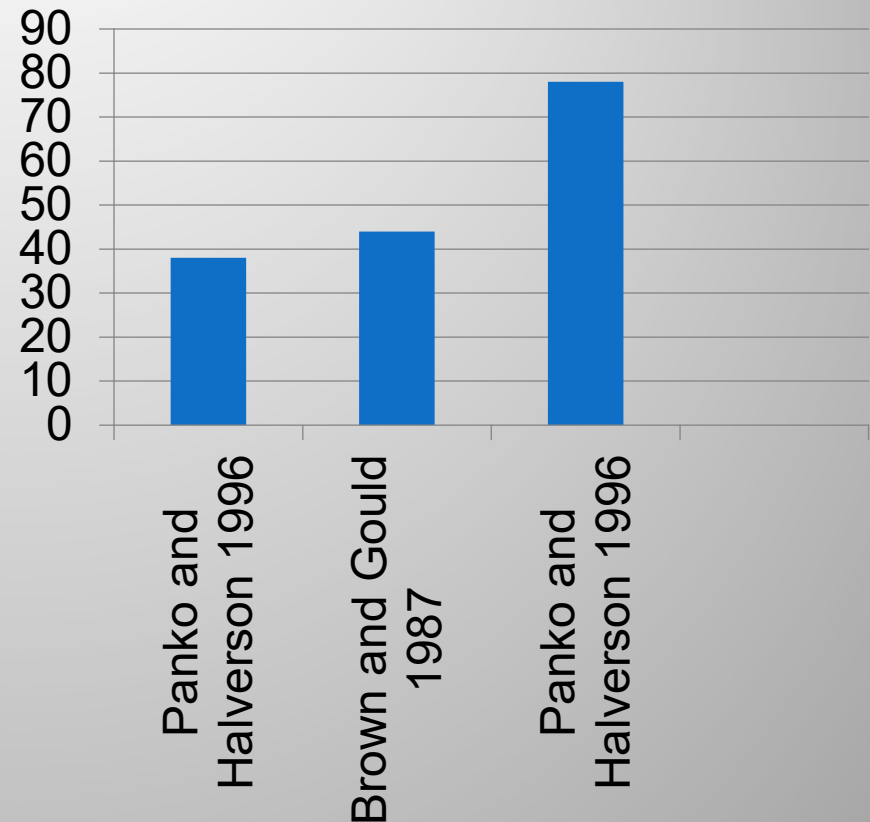
Based on: Eight of the Worst Spreadsheet Blunders, Thomas Wailgum, August 17, 2007
http://www.cio.com/article/131500/Eight_of_the_Worst_Spreadsheet_Blunders?page=1&taxonomyId=3000

How Many Spreadsheets?



Erroneous Spreadsheets

- 44% of “finished” spreadsheets still had errors [Brown and Gould 1987]
- other such studies reported errors in 38% to 77% of spreadsheets at a similar stage [Panko and Halverson 1996]



Spreadsheet Error Density

Figure 3: Audits of Real-World Spreadsheets

Authors	Year	Number of SSs Audited	Average Size (Cells)	Percent of SSs with Errors	Cell Error Rate	Comment
Hicks	1995	1	3,856	100%	1.2%	One omission error would have caused an error of more than a billion dollars.
Coopers & Lybrand	1997	23	More than 150 rows	91%		Off by at least 5%. This amount could indicate
KPMG	1998	22		91%		Only significant errors that could affect decisions.
Lukasic	1998	2	2,270 & 7,027	100%	2.2%, 2.5%	In Model 2, the investment's value was overstated by 16%. Quite serious.
Butler	2000	7		86%	0.4%**	Only errors large enough to require additional tax payments.**
Clermont, Hanin, & Mittermeier	2002	3		100%	1.3%, 6.7%, 0.1%	Computed on the basis of non-empty cells.
Interview I*	2003	~36 / yr		100%		Approximately 5% had <i>extremely</i> serious errors.
Interview II*	2003	~36 / yr		100%		Approximately 5% had <i>extremely</i> serious errors.
Lawrence and Lee	2004	30	2,182 unique formulas	100%	6.9%	30 most financially significant SSs audited by Mercer Finance & Risk Consulting in previous year.
Total		88		94%	5.2%	

Raymond R. Panko, University of Hawai'i

Imperative Software Error Rates

Application Domain	Number Projects	Error Range (Errors/KESLOC)	Normative Error Rate (Errors/KESLOC)	Notes	
Automation	55	2 to 8	5	Factory automation	
Banking	30	3 to 10	6	Loan processing, ATM	5
Command & Control	45	0.5 to 5	1	Command centers	6
Data Processing	35	2 to 14	8	DB-intensive systems	1
Environment/Tools	75	5 to 12	8	CASE, compilers, etc.	8
Military -All	125	0.2 to 3	< 1.0	See subcategories	8
§ Airborne	40	0.2 to 1.3	0.5	Embedded sensors	< 1.0
§ Ground	52	0.5 to 4	0.8	Combat center	0.5
§ Missile	15	0.3 to 1.5	0.5	GNC system	0.8
§ Space	18	0.2 to 0.8	0.4	Attitude control system	0.5
Scientific	35	0.9 to 5	2	Seismic processing	0.4
Telecommunications	50	3 to 12	6	Digital switches	2
Test	35	3 to 15	7	Test equipment, devices	6
Trainers/Simulations	25	2 to 11	6	Virtual reality simulator	7
Web Business	65	4 to 18	11	Client/server sites	6
Other	25	2 to 15	7	All others	11
					7
					4.613333

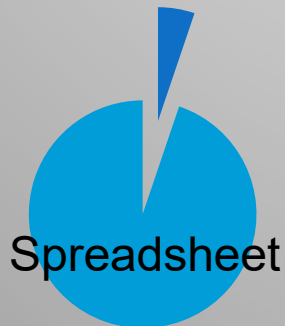
Donald Reifer, "Industry Software Cost, Quality, and Productivity Benchmarks", DoD Software Tech News, July 2004

Comparison Spreadsheet to Software Error Rates

Spreadsheets – average audited
5.2% error rate

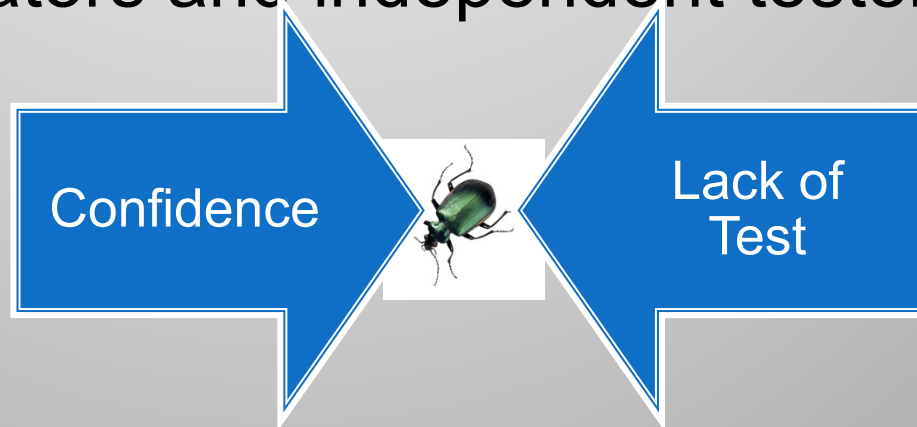
Software – 4.6 per KSLOC or
.46% error rate

So spreadsheets 10 times more
likely to have errors than software.



Reason For Errors

- The unwarranted confidence creators of spreadsheets seem to have in the reliability of those spreadsheets [Wilcox et al. 1997]
- Testing by creators and independent testers is not common.

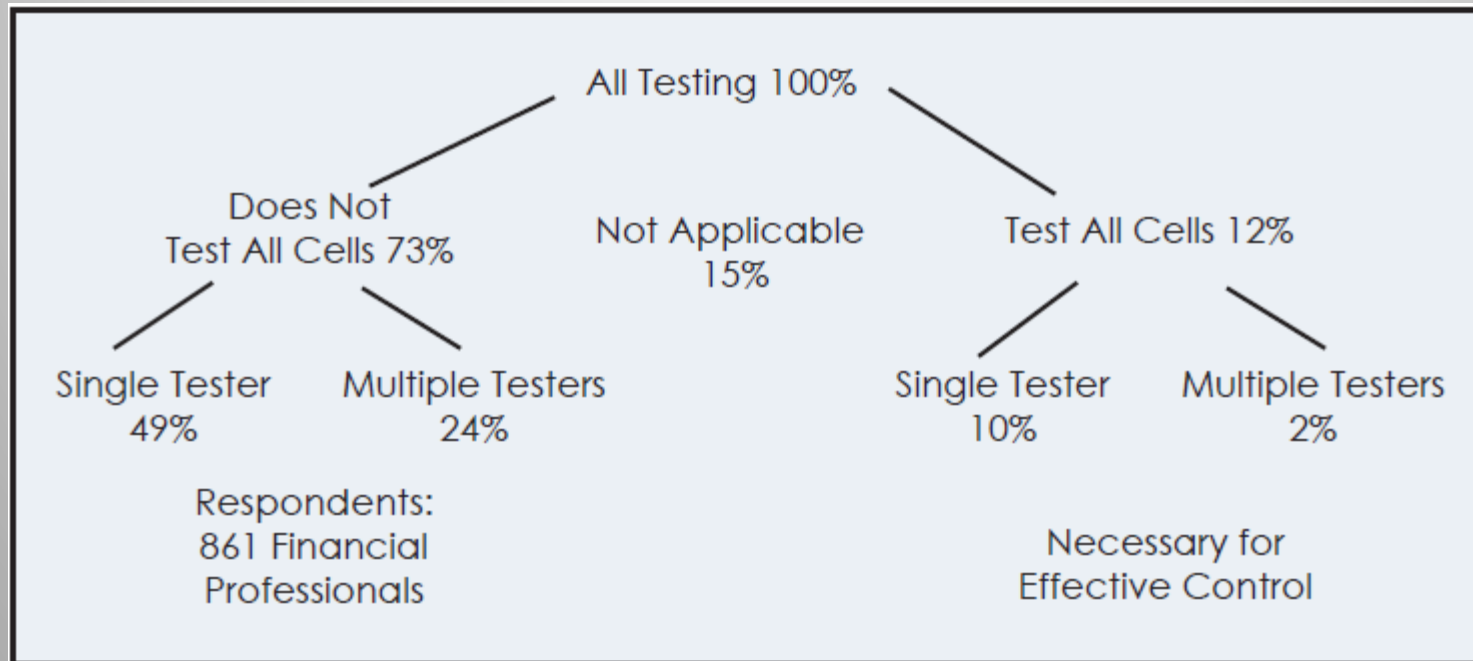


Most Companies Do Not Test Spreadsheets

Almost None	24%
Under 10%	20%
11% to 25%	12%
Over 25%	17%
Nearly All	16%
Not Applicable	11%
Total	100%
Respondents	862

Panko, Raymond R. (2005, July 7/8), "Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting," *EuSpRIG 2005*, University of Greenwich, London, UK. European Spreadsheet Research Information Group. <http://www.eusprig.org>.

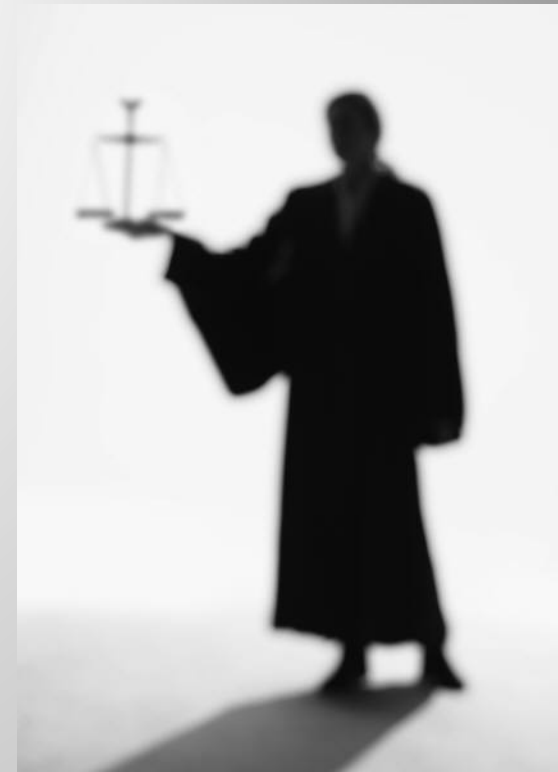
Independent Spreadsheet Testing Not Commonplace



Panko, Raymond R. (2005, July 7/8), "Sarbanes–Oxley: What about All the Spreadsheets? Controlling for Errors and Fraud in Financial Reporting," *EuSpRIG 2005*, University of Greenwich, London, UK. European Spreadsheet Research Information Group. <http://www.eusprig.org>.

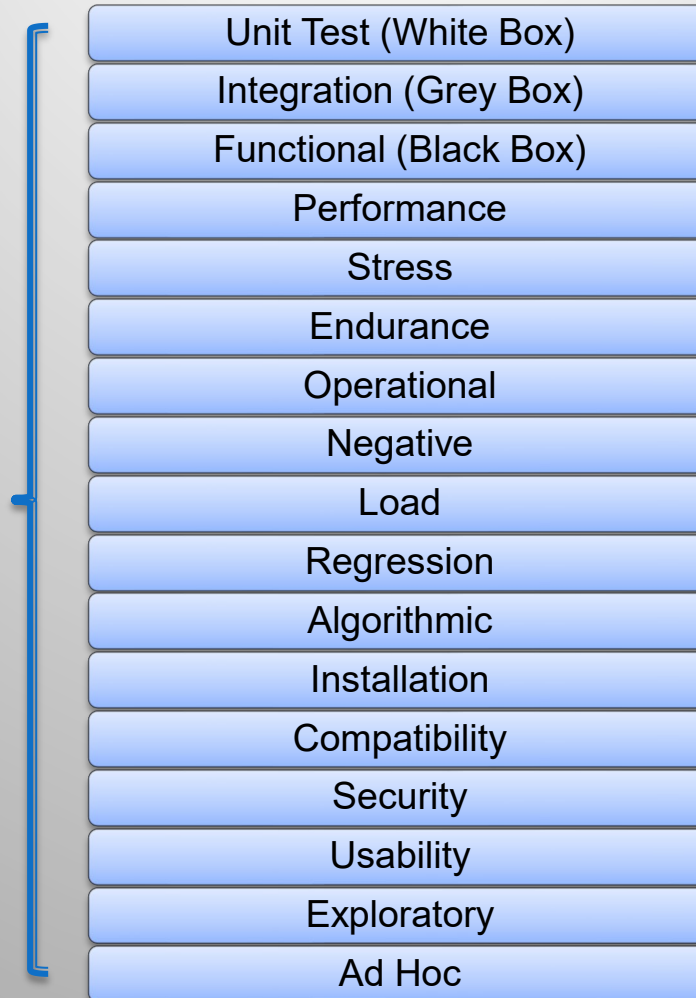
Sarbanes-Oxley Act (SOX)

- US Federal law enacted in July 30, 2002
- Named after Paul Sarbanes (D-Md) and Michael G. Oxley (R-Oh)
- Largely adopted in Canada but less prescriptive
- Top management must now individually certify the accuracy of financial information.



Lots of Imperative Programming Testing Techniques

Imperative Programming



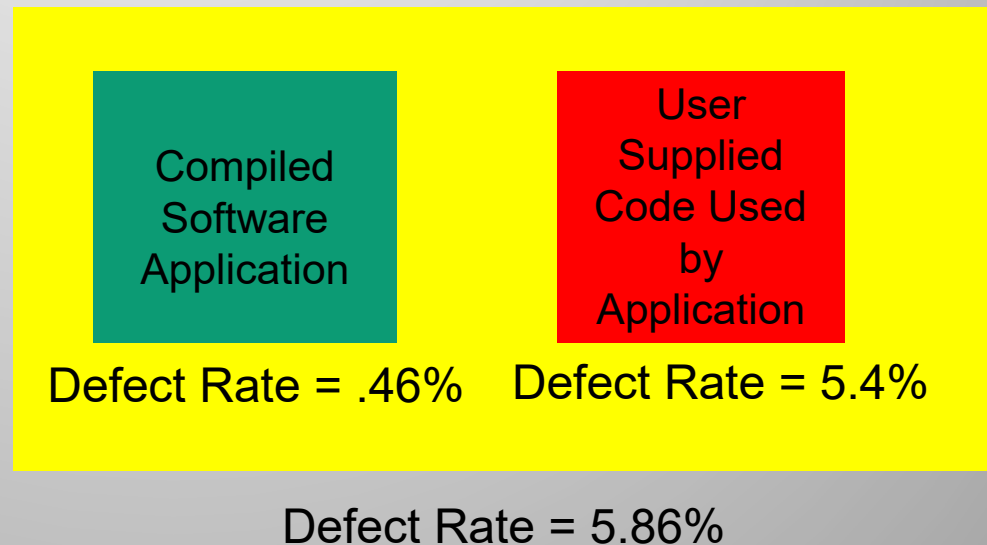
Lots of Progress on When to Test

Imperative
Programming



Quality of Second Order Software Matters:

- Spreadsheets
- R (Statistical Modeler)
- Aspen
- LabView
- MATLAB
- Mathematica
- Python Steering

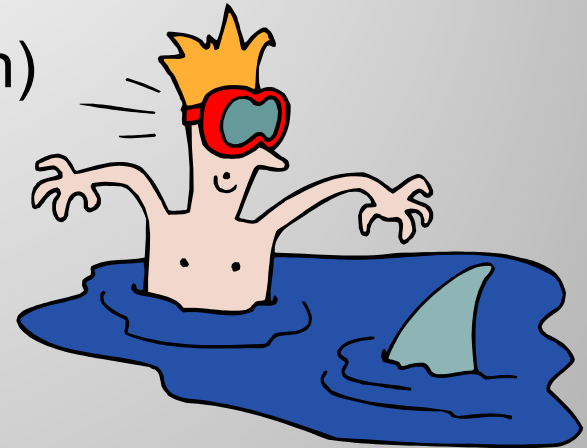


Differences Between Spreadsheets and Software

- Evaluation Order
- Most data references statically resolved
- No loops (ignoring macros)
- Incremental visual feedback
- Automatic recalculation
- User base not skilled in programming and testing

Testing Spreadsheet Risk

- Oracle for calculations
 - Check another source (validation)
 - Can be calculated in parallel
- Test has bugs
- Test Coverage
- Spreadsheet requirements scarce
- Adds Time



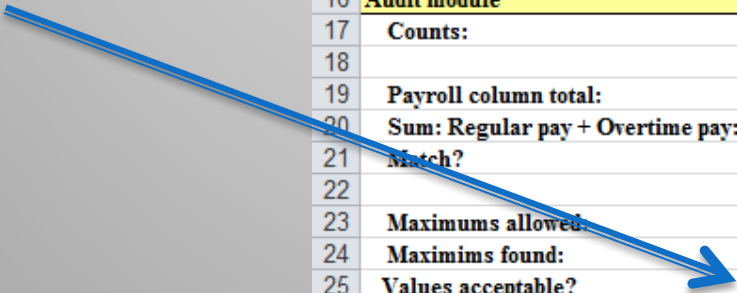
Spreadsheet Self Tests

	A	B	C	D	E	F	G
1	Choi Construction Company						
2	Payroll for week ending: 3/15/XX						
3							
4		Payrate	Regular hours	Overtime hours	Regular pay	Overtime pay	Payroll totals
5	Adams	8.90	40	3	356.00	40.05	396.05
6	Baker	12.55	35	0	502.00	0.00	502.00
7	Carlton	9.60	40	2	384.00	38.80	422.80
8	Daniels	10.20	35	0	408.00	0.00	408.00
9	Englert	9.60	40	5	384.00	72.00	456.00
10	Franklin	11.55	40	2	462.00	34.65	496.65
11	Griffin	10.80	35	0	432.00	0.00	432.00
12	Hartford	9.90	40	9	396.00	133.65	529.65
13							
14	Totals:		305	21	\$ 3,324.00	\$ 319.15	\$ 3,643.15
15							
16	Audit module						
17	Counts:		8	5			
18							
19	Payroll column total:						\$ 3,643.15
20	Sum: Regular pay + Overtime pay:						3,643.15
21	Match?						Yes
22							
23	Maximums allowed:	\$ 14.00	40.00	10.00	\$ 560.00		
24	Maximums found:	\$ 12.55	40.00	9.00	\$ 502.00		
25	Values acceptable?	Yes	Yes	Yes	Yes		

B24=MAX(B5:B12)



B25 =IF(B24<=B23, "Yes", "No")



Ferret Out Spreadsheet Errors, Mark Simon 2004, Journal of Accountancy

Parallel Copy Technique

Exhibit 1

	A	B	C	D	E	F	G
1	Choi Construction Company						
2	Payroll for Week Ending: 3/15/XX						
3							
4		Payrate	Regular Hours	Overtime Hours	Regular Pay	Overtime Pay	Payroll Totals
5	Adams	8.90	40	3	356.00	40.05	396.05
6	Baker	12.55	35	0	502.00	0.00	502.00
7	Carlton	9.60	40	2	384.00	38.80	422.80
8	Daniels	10.20	35	0	408.00	0.00	408.00
9	Englert	9.60	40	5	384.00	72.00	456.00
10	Franklin	11.55	40	2	462.00	34.65	496.65
11	Griffin	10.80	35	0	432.00	0.00	432.00
12	Hartford	9.90	40	9	396.00	133.65	529.65
13							
14	Totals:		305	21	\$ 3,324.00	\$ 319.15	\$ 3,643.15
15							
16	Audit Module						
17	Counts:		8	5			
18							
19	Payroll Column Total:					\$ 3,643.15	
20	Sum: Regular Pay + Overtime Pay:					3,643.15	
21	Match?					Yes	
22							
23	Maximums Allowed:	\$ 14.00	40.00	10.00	\$ 560.00		
24	Maximums Found:	\$ 12.55	40.00	9.00	\$ 502.00		
25	Values acceptable?	Yes	Yes	Yes	Yes		
26							
27							
28	(Signature)						
29	I have examined this spreadsheet and						
30	vouch for its accuracy and completeness.						

Exhibit 1

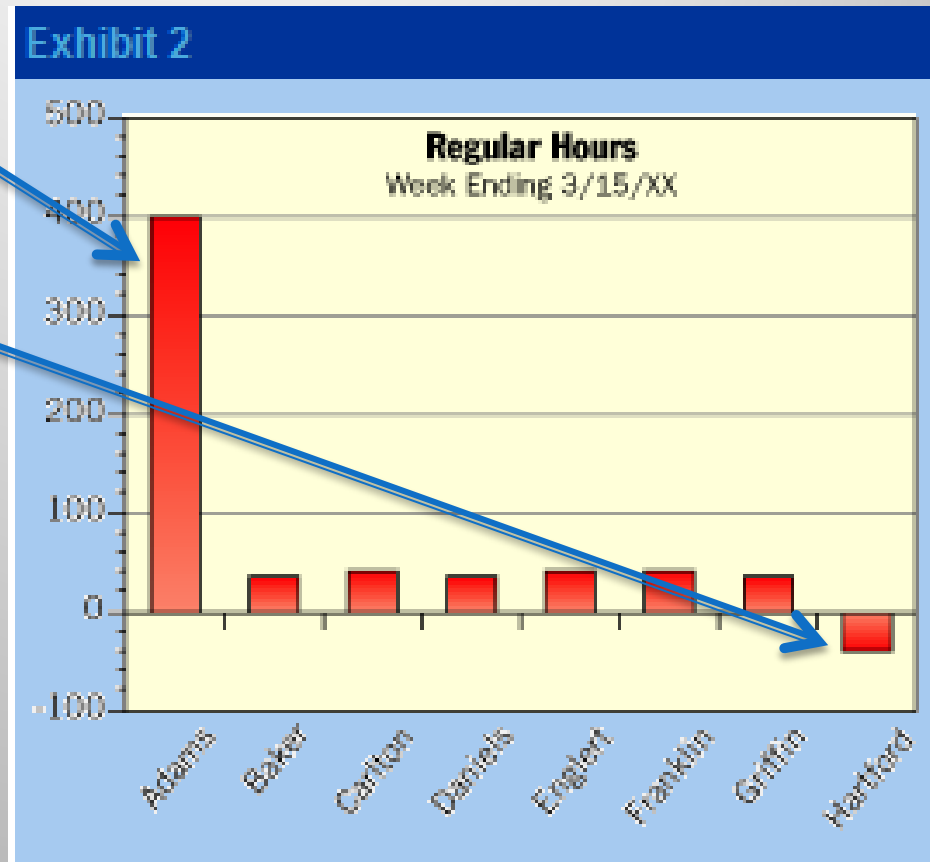
	A	B	C	D	E	F	G
1	Choi Construction Company						
2	Payroll for Week Ending: 3/15/XX						
3							
4		Payrate	Regular Hours	Overtime Hours	Regular Pay	Overtime Pay	Payroll Totals
5	Adams	8.90	40	3	356.00	40.05	396.05
6	Baker	12.55	35	0	502.00	0.00	502.00
7	Carlton	9.60	40	2	384.00	38.80	422.80
8	Daniels	10.20	35	0	408.00	0.00	408.00
9	Englert	9.60	40	5	384.00	72.00	456.00
10	Franklin	11.55	40	2	462.00	34.65	496.65
11	Griffin	10.80	35	0	432.00	0.00	432.00
12	Hartford	9.90	40	9	396.00	133.65	529.65
13							
14	Totals:		305	21	\$ 3,324.00	\$ 319.15	\$ 3,643.15
15							
16	Audit Module						
17	Counts:		8	5			
18							
19	Payroll Column Total:					\$ 3,643.15	
20	Sum: Regular Pay + Overtime Pay:					3,643.15	
21	Match?					Yes	
22							
23	Maximums Allowed:	\$ 14.00	40.00	10.00	\$ 560.00		
24	Maximums Found:	\$ 12.55	40.00	9.00	\$ 502.00		
25	Values acceptable?	Yes	Yes	Yes	Yes		
26							
27							
28	(Signature)						
29	I have examined this spreadsheet and						
30	vouch for its accuracy and completeness.						



Ferret Out Spreadsheet Errors, Mark Simon 2004, Journal of Accountancy

Plot Values, Check for Patterns

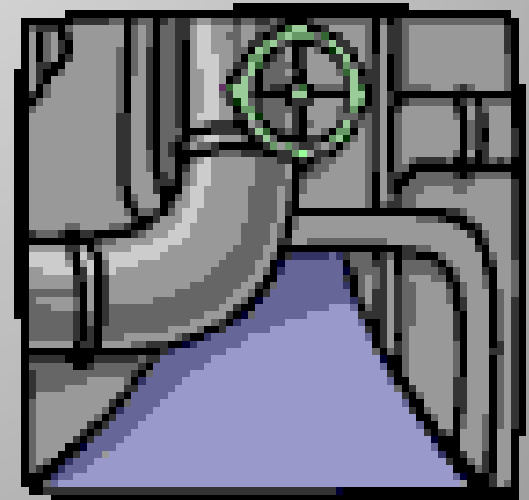
- Too many zeros
- Sign flip

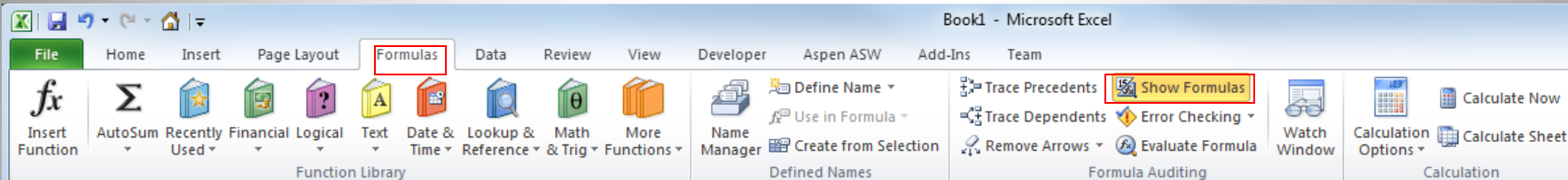


Ferret Out Spreadsheet Errors, Mark Simon 2004, Journal of Accountancy

Sensitivity Analysis

- Manipulate inputs
- Predict how other cells should change
- Check that they change as expected
- Example
 - Pay Rate goes up
 - Total pay goes up





Formula View

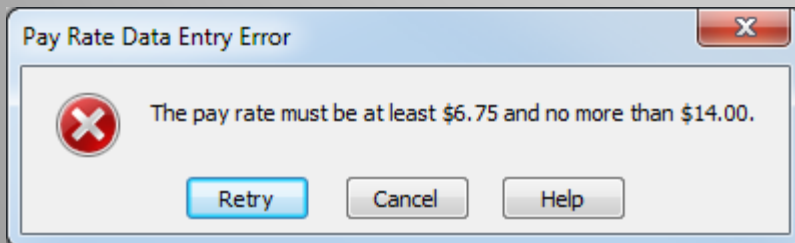
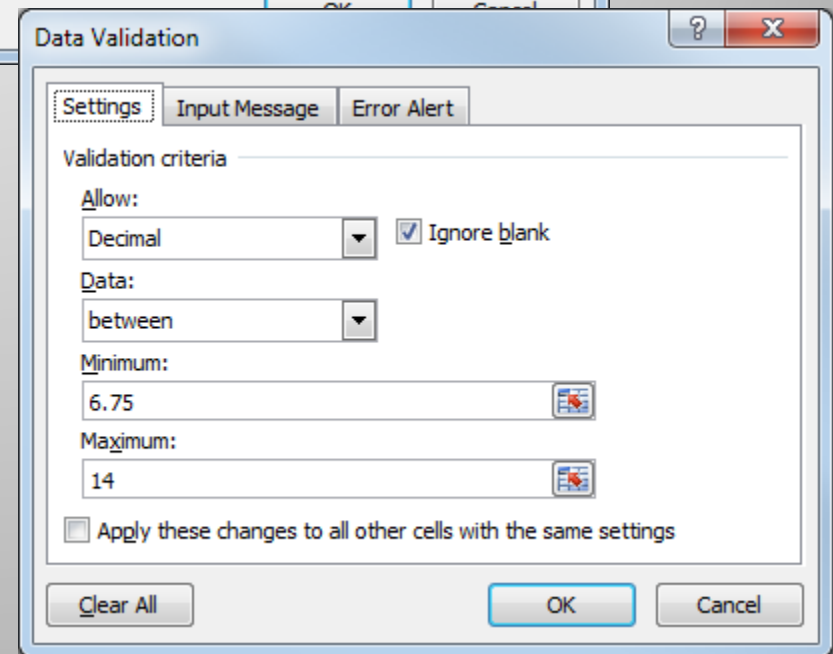
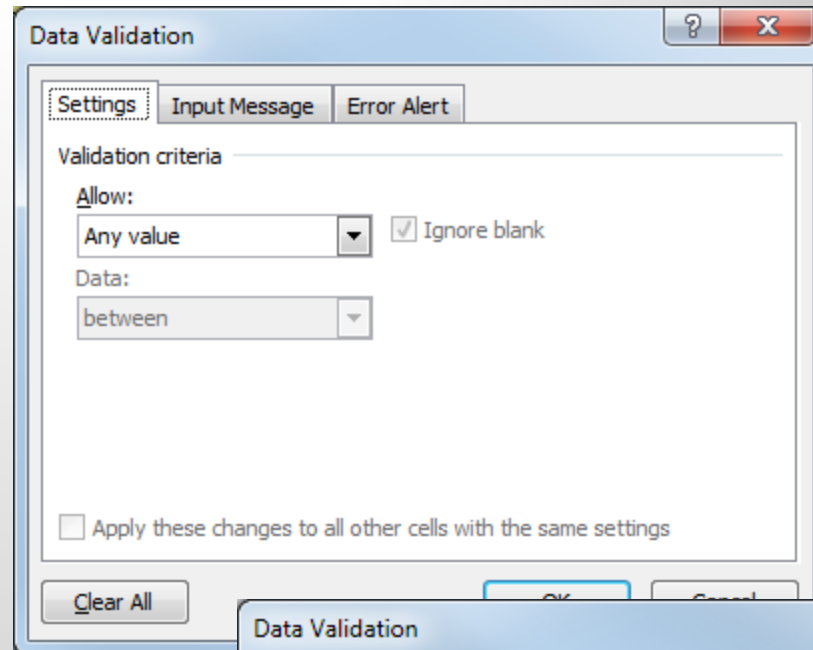
- Tools/Options/View/check formulas or CTRL~
- Not filled down
- Constant

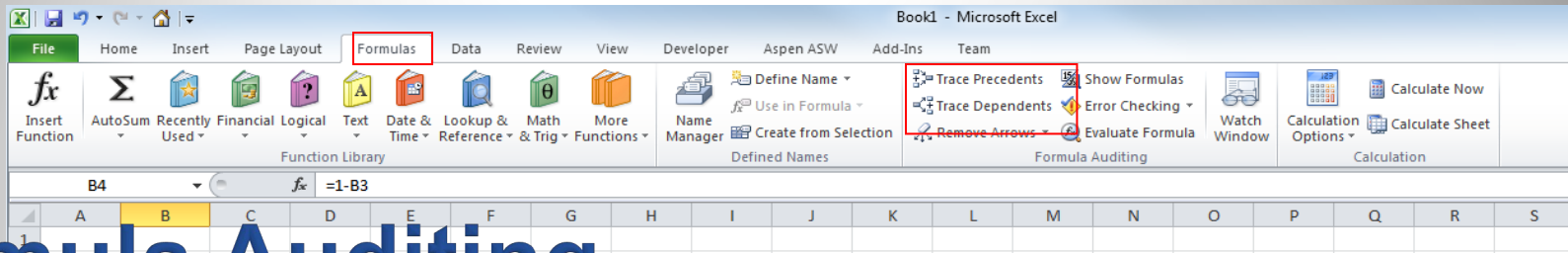
Regular pay	Overtime pay	Payroll totals
=B5*C5	=B5*1.5*D5	=E5+F5
=B6*40	=B6*1.5*D6	=E6+F6
=B7*40	38.8	=E7+F7
=B8*40	=B8*1.5*D8	=E8+F8
=B9*40	=B9*1.5*D9	=E9+F9
=B10*40	=B10*1.5*D10	=E10+F10
=B11*40	=B11*1.5*D11	=E11+F11
=B12*40	=B12*1.5*D12	=E12+F12
=SUM(E5:E12)	=SUM(F5:F12)	=SUM(G5:G12)

Ferret Out Spreadsheet Errors, Mark Simon 2004, Journal of Accountancy

Data Validation

- Select cells
- Data/Data Validation
- Put \$5 in cell B5



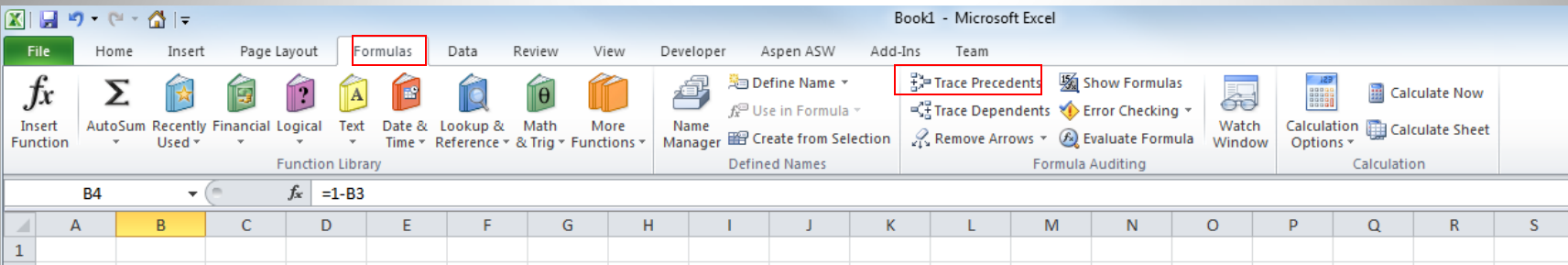


Formula Auditing

- Trace Precedents
- Trace Dependents

Ferret Out Spreadsheet Errors,
Mark Simon 2004, Journal of
Accountancy

Choi Construction Company						
Payroll for week ending: 3/15/XX						
	Payrate	Regular hours	Overtime hours	Regular pay	Overtime pay	Payroll totals
Adams	8.90	40	3	356.00	40.05	396.05
Baker	12.55	35	0	502.00	0.00	502.00
Choi Construction Company						
Payroll for week ending: 3/15/XX						
	Payrate	Regular hours	Overtime hours	Regular pay	Overtime pay	Payroll totals
Adams	8.90	40	3	356.00	40.05	396.05
Baker	12.55	35	0	502.00	0.00	502.00
Carlton	9.60	40	2	384.00	38.80	422.80
Daniels	10.20	35	0	408.00	0.00	408.00
Englert	9.60	40	5	384.00	72.00	456.00
Franklin	11.55	40	2	462.00	34.65	496.65
Griffin	10.80	35	0	432.00	0.00	432.00
Hartford	9.90	40	9	396.00	133.65	529.65
Totals:		305	21	\$ 3,324.00	\$ 319.15	\$ 3,643.15
Audit module						
Counts:		8	5			
Payroll column total:						\$ 3,643.15
Sum: Regular pay + Overtime pay:						3,643.15
Match?						Yes
Maximums allowed:	\$ 14.00	40.00	10.00	\$ 560.00		
Maximums found:	\$ 12.55	40.00	9.00	\$ 502.00		
Values acceptable?	Yes	Yes	Yes	Yes		



Formula Auditing

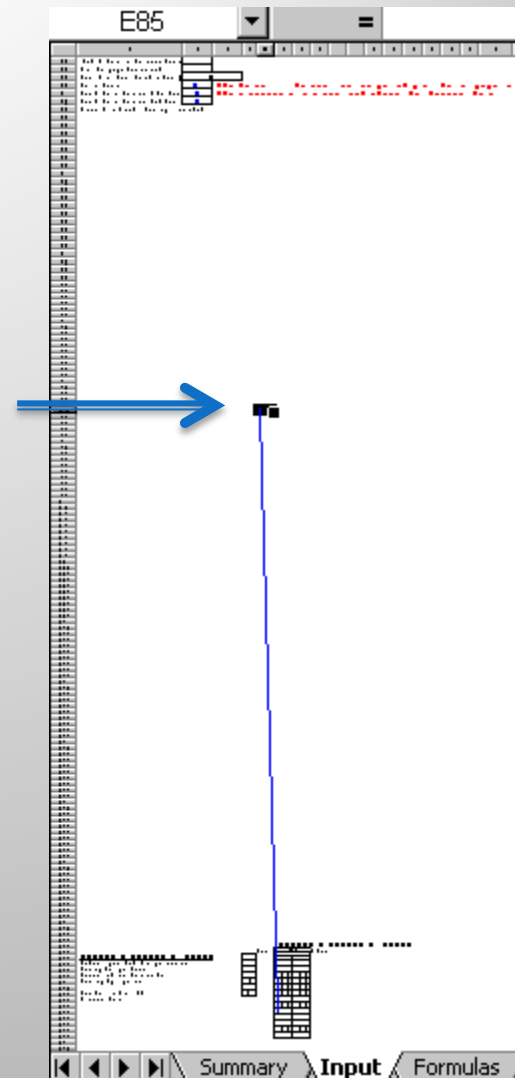
- Multiple selection of precedents
- Inconsistent dot patterns show errors

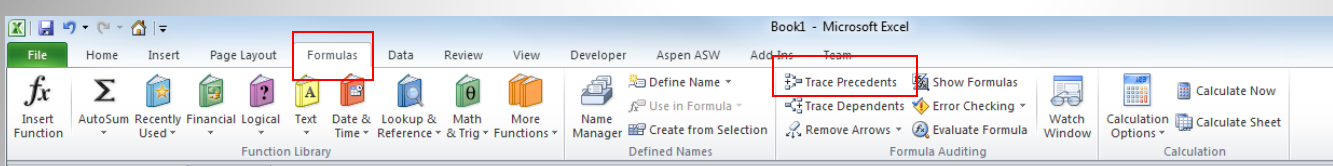
Choi Construction Company						
Payroll for week ending: 3/15/XX						
	Payrate	Regular hours	Overtime hours	Regular pay	Overtime pay	Payroll totals
Adams	8.90	40	3	356.00	40.05	396.05
Baker	12.55	35	0	502.00	0.00	502.00
Carlton	9.60	40	2	384.00	38.80	422.80
Daniels	10.20	35	0	408.00	0.00	408.00
Englert	9.60	40	5	384.00	72.00	456.00
Franklin	11.55	40	2	462.00	34.65	496.65
Griffin	10.80	35	0	432.00	0.00	432.00
Hartford	9.90	40	9	396.00	133.65	529.65
Totals:		305	21	\$ 3,324.00	\$ 319.15	\$ 3,643.15
Audit module						
Counts:		8	5			
Payroll column total:						\$ 3,643.15
Sum: Regular pay + Overtime pay:						3,643.15
Match?						Yes
Maximums allowed:	\$ 14.00	40.00	10.00	\$ 560.00		
Maximums found:	\$ 12.55	40.00	9.00	\$ 502.00		
Values acceptable?	Yes	Yes	Yes	Yes		

Ferret Out Spreadsheet Errors, Mark Simon 2004, Journal of Accountancy

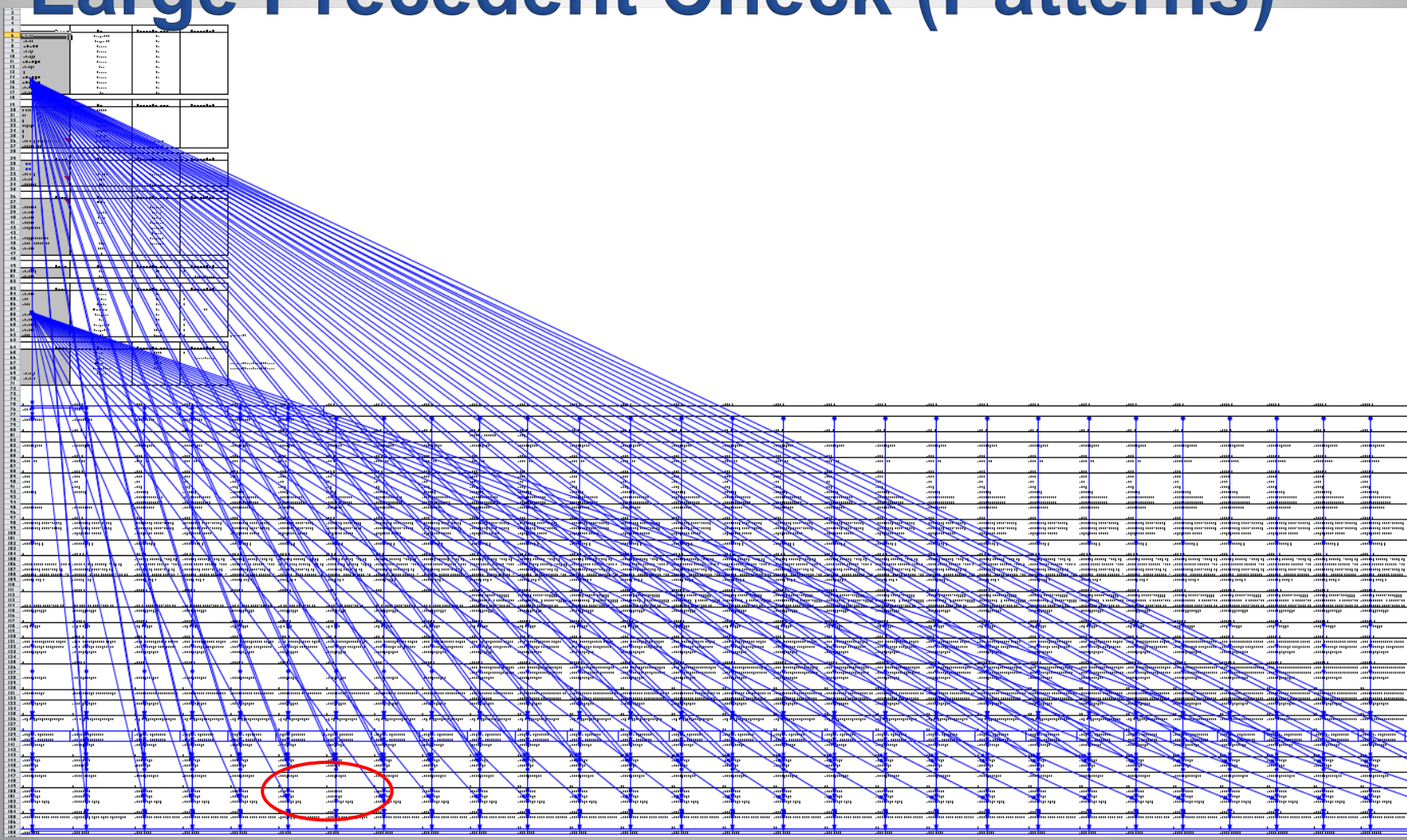
Formula Auditing

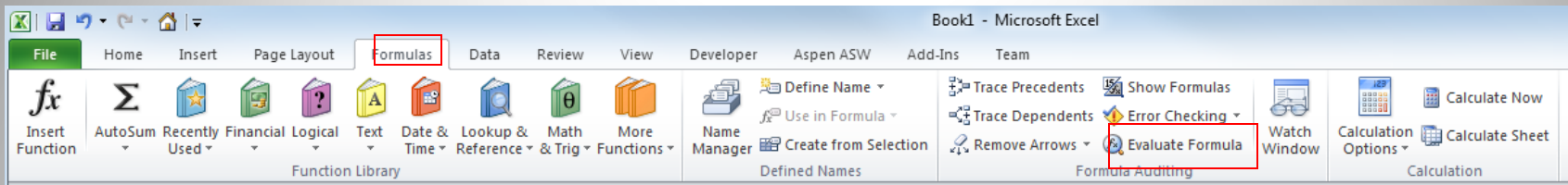
Formula references a blank cell in sparse area



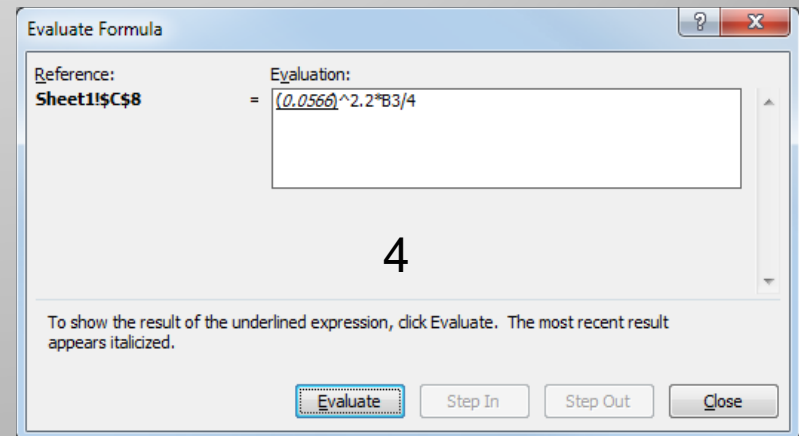
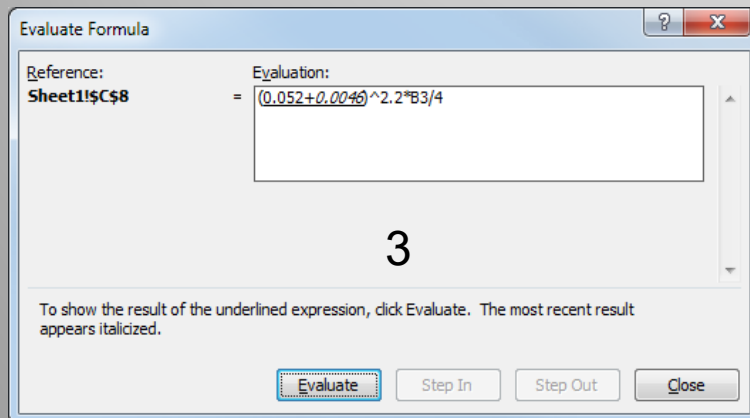
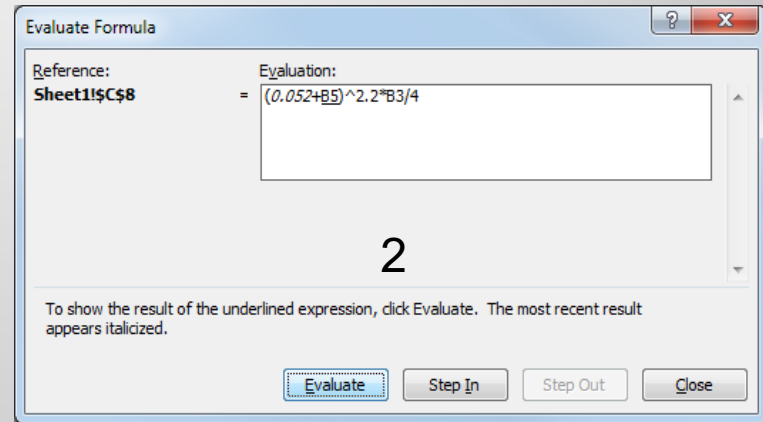
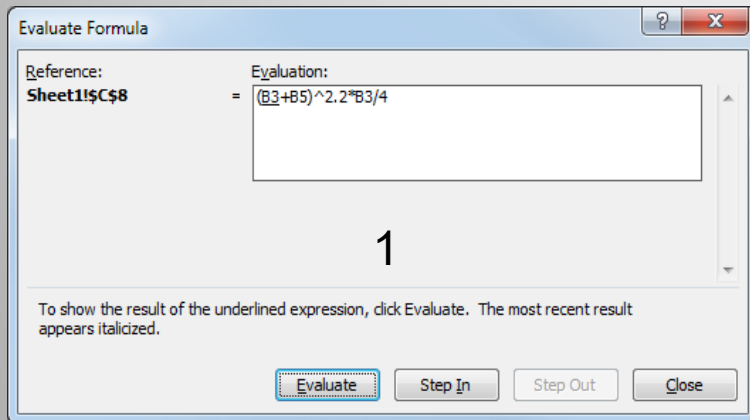


Large Precedent Check (Patterns)





Evaluate Formula



Static Error Checks

Start every function with the equal sign (=)

Match all open and close parentheses

Use a colon to indicate a range

Enter all required arguments

Enter the correct type of arguments

Nest no more than 64 functions

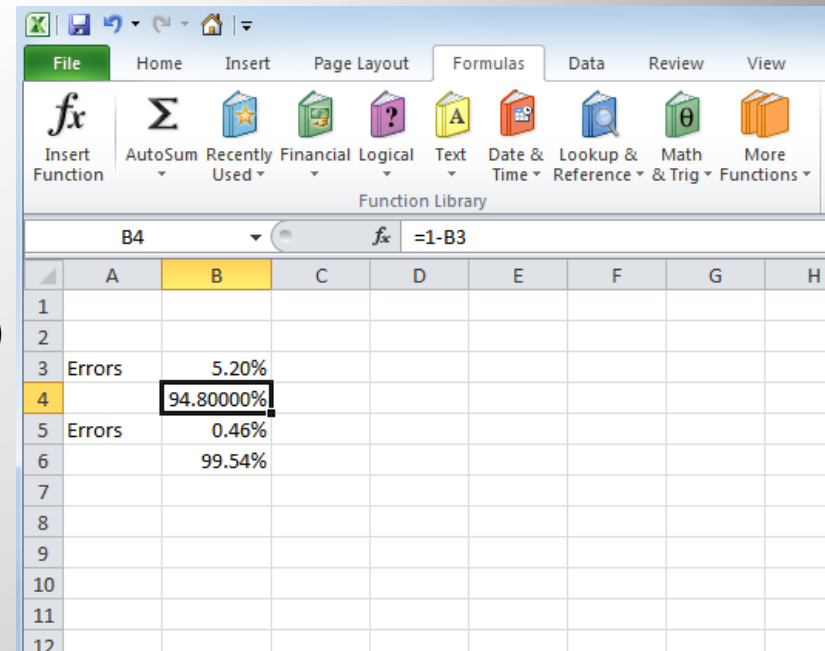
Enclose other sheet names in single quotation marks

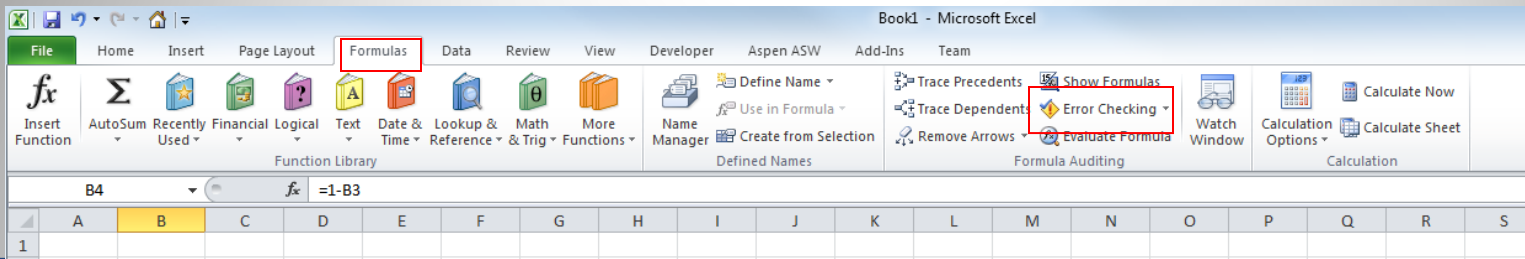
Place an exclamation point (!) after a worksheet name when you refer to it in a formula

Include the path to external workbooks

Enter numbers without formatting

Avoid dividing by zero





Error Checking (Formula/Error Checking)

- Cells containing formulas that result in an error
- Inconsistent calculated column formula in tables
- Cells containing years represented as 2 digits
- Numbers formatted as text or preceded by an apostrophe
- Formulas inconsistent with other formulas in the region
- Formulas which omit cells in a region
- Unlocked cells containing formulas (Review/Unprotect)
- Formulas referring to empty cells
- Data entered in a table is invalid

Accountability



(Signature)

**I have examined this spreadsheet and
vouch for its accuracy and completeness.**

Using VB Macros to Test Spreadsheet

- Read and write cells
- Manipulate objects
- Boundary tests
- Random tests
- Tests stay with spreadsheet
- **Example**

The screenshot shows a Microsoft Excel spreadsheet titled "Software Grading Tool w tests.xlsx - Microsoft Excel". The spreadsheet has columns A through W and rows 1 through 32. The data is organized into a table with the following structure:

Consequence of Failure Tiers	Risk Grade		
Tier 0	RL1	RL1	RL1
Tier 1	RL3	RL2	RL2
Tier 2	RL4	RL3	RL3
Tier 3	RL5	RL4	RL3
Tier 4	RL5	RL5	RL5

Below the table, there are arrows indicating a range from column 2 to column 8, and a label "Process / Development-Environment".

The VBA macro editor shows the following code:

```

Sub Random_test()
    ' Test random values for all selections
    ' Gregory Epc
    ' 12/15/2013
    ' LLNL
    'Random_test Macro
    For Index = 1 To 1000
        Sheets("Application Characteristics").Select
        Range("E3").Select
        ActiveCell.FormulaR1C1 = "Test Random"
        Range("G5").Select
    ' Select non B30
        ActiveCell.FormulaR1C1 = "2"
        Range("G7").Select
    ' Select major control
        ActiveCell.FormulaR1C1 = "3"
        Range("I10:E10").Select
        ActiveCell.FormulaR1C1 = "This is the random test"
        Sheets("Risk Consequences").Select
    ' Set consequences of failure
        Range("E17").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Range("C17").Select
        ActiveCell.FormulaR1C1 = Int((4 - 1 + 1) * Rnd + 1)
        Range("D17").Select
        ActiveCell.FormulaR1C1 = Int((4 - 1 + 1) * Rnd + 1)
        Range("E17").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Sheets("Process/Development Risks").Select
        Range("I3").Select
    ' Set Programming Development Environment
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Range("I9").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Range("L7").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Range("I9").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
        Range("L11").Select
        ActiveCell.FormulaR1C1 = Int((6 - 1 + 1) * Rnd + 1)
    End For
End Sub
    
```

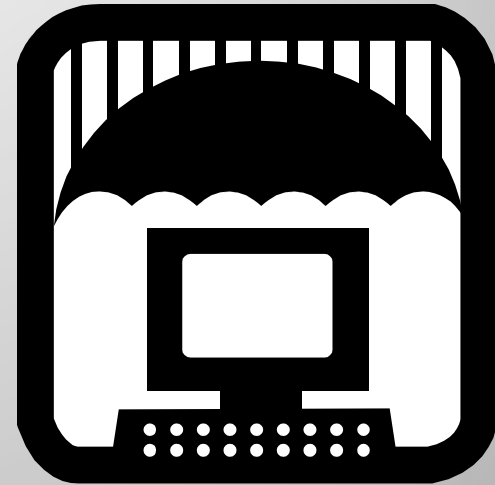
At the bottom of the spreadsheet, there is a section titled "Self Tests" with the following entries:

Self Tests	Expected Result
Min	Lower left RL5 selected
Max	Upper right RL1 selected
Half	Middle RL3 selected
Valid	Five steps from lower left to upper right
Random	RL level jumps around 25 times
830	Check that min for 830 mode is RL4
Minor	Check that Process Maturity, Schedule & Resource Constraints, Risk Resolution, Personnel Capability, Team Dynamics are not used, Organization Reputation used.
None	Check that only Product Volatility, Software Complexity, Degree of Innovation, Software Size, Organizational Reputation is active

Example sheet available at

Test Coverage

- Tests should cover each function
- Boundary tests
 - Minimum +/- 1
 - Maximum +/-1
 - Half Scale
 - Valid Range
- Stress
 - Very large or small numbers
 - Invalid values (negative, mixed type, etc.)
- Negative (if possible)
- Random



Testing Tests

- Error seed spreadsheet
 - Add +1 or -1 to answer
 - Flip sign
 - Change cell reference
- Run Tests
- Error should be detected
- Tests should fail
- Seeding/Clearing automated to prevent errors
- Indicator that error is in spreadsheet

The screenshot displays an Excel spreadsheet with the following content:

Consequence of Failure Tiers	Risk Grade		
Tier 0	RL1	RL1	RL1
Tier 1	RL3	RL2	RL2
Tier 2	RL4	RL3	RL3
Tier 3	RL5	RL4	RL3
Tier 4	RL5	RL5	RL5

Below the table, there are instructions: "If rating within 20% of next higher boundary, use professional judgment to decide if a higher Risk Level grade is warranted." and "Push each self test button, verify expected results".

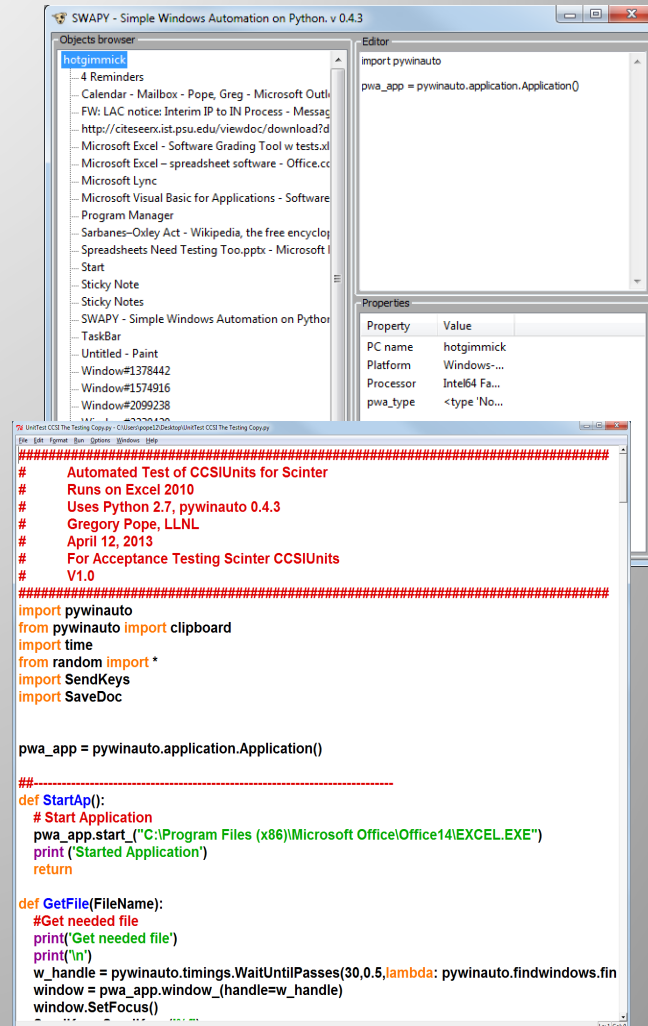
The 'Self Tests' section includes buttons for: Min, Max, Half, Valid, Random, 830, Minor, None, and Run All. Each button has a corresponding description of the test's purpose.

The 'Bug Seeding' section includes buttons for: Tier Bug, PDE Bug, 830 Bug, Minor Bug, and None Bug. Each button has a 'Clear' button and a description of the bug's effect.

Using Pywinauto and SWAPY

- Independent of Spreadsheet
- Windows native application
- Combined with Ghost Mouse
- SWAPY reads window objects
- Include pywinauto to python

Example



The screenshot shows the SWAPY application interface. The title bar reads "SWAPY - Simple Windows Automation on Python. v 0.4.3". The interface is divided into three main sections:

- Objects browser:** A tree view showing a list of windows. The selected window is "hotgimmick". Other visible windows include "4 Reminders", "Calendar - Mailbox - Pope, Greg - Microsoft Outl...", "FW: LAC notice: Interim IP to IN Process - Messag...", "http://cite.esci.ist.psu.edu/viewdoc/download?d...", "Microsoft Excel - Software Grading Tool w tests.xl...", "Microsoft Excel - spreadsheet software - Office.cc...", "Microsoft Lync", "Microsoft Visual Basic for Applications - Software...", "Program Manager", "Sarbanes-Oxley Act - Wikipedia, the free encyclo...", "Spreadsheets Need Testing Too.pptx - Microsoft I...", "Start", "Sticky Note", "Sticky Notes", "SWAPY - Simple Windows Automation on Python", "TaskBar", "Untitled - Paint", "Window#1378442", "Window#1574916", and "Window#2099238".
- Editor:** A text editor containing the following Python code:

```
import pywinauto
pwa_app = pywinauto.application.Application()
```
- Properties:** A table showing properties for the selected window:

Property	Value
PC name	hotgimmick
Platform	Windows-...
Processor	Intel64 Fa...
pwa_type	<type 'No...

Below the SWAPY window, a separate window displays the following Python code for an automated test:

```
#####
# Automated Test of CCSUnits for Scinter
# Runs on Excel 2010
# Uses Python 2.7, pywinauto 0.4.3
# Gregory Pope, LLNL
# April 12, 2013
# For Acceptance Testing Scinter CCSUnits
# V1.0
#####
import pywinauto
from pywinauto import clipboard
import time
from random import *
import SendKeys
import SaveDoc

pwa_app = pywinauto.application.Application()

##-----
def StartAp():
    # Start Application
    pwa_app.start('C:\Program Files (x86)\Microsoft Office\Office14\EXCEL.EXE')
    print('Started Application')
    return

def GetFile(FileName):
    #Get needed file
    print('Get needed file')
    print('\n')
    w_handle = pywinauto.timings.WaitUntilPasses(30,0.5,lambda: pywinauto.findwindows.find
    window = pwa_app.window(handle=w_handle)
    window.SetFocus()
```

Conclusion

- There is much known about how to prevent and detect spreadsheet errors.
- There is much known about how to test software
- The two knowledge bases have yet to merge
- We do not do a good job of testing spreadsheets
- Let us merge these areas starting today, eh?

Further Reading: *Spreadsheet Check and Control 47 key practices to detect and prevent errors*,
Patrick O'Beirne