Intermediate/Advanced Excel, and VBA, for Real Estate Finance

Stephen Malpezzi
smalpezzi@bus.wisc.edu
http://www.bus.wisc.edu/realestate

"Tom Tomorrow"
(A few years ago...)

Key Spreadsheets

- "Simple Excel Proforma.xls"
  - A "toy" model, easy to navigate, demonstrates some principles of good design.
- "Simple Proforma with Detailed Development and Scenarios.xls"
  - Multiple worksheet design, Scenario Manager
- "Apartment Proforma 1.xls"
  - Includes use of IF and ISBLANK functions for flexible input assumptions.

Objectives

- Examine several proformas in Excel.
  - A very simple model
  - Other models if we have time
- Focus on good design, comments, auditing, named ranges
- Intro to VBA
- VBA Project: Proforma Model for Multiple Cases

History of Spreadsheets

- Visicalc
  - The mother of all spreadsheets. (Imagine life without one). Invented by two B-School students to save time on homework. The "killer app" for the Apple II.
- Lotus 1-2-3
  - Macros! Functions! Graphs! 95% Market Share!
- Quattro
  - Copyable! Decent (?) graphing and formatting, macro recording.
- Quattro Pro, Lotus v. 3+, Supercalc, Excel
  - Mouse support, multiple pages, link to other applications, more functions, better graphics.
- Excel 97, 2000, 2002, 2003
  - Market leader. Full-blown programming language (VBA) built in. Good integration with other MS products.

History of Spreadsheets: Excel 2007

- Better graphics, new fonts
- The Ribbon.
  - Works better in the long run; better groupings, more flexible. But a pain to get used to.
  - Ease of adding oft-used commands to the "Quick Access Toolbar" is nice.
- New file formats! Extensions:
  - xls is our familiar Excel spreadsheet
  - xlsx denotes 2007 format
  - xlsm denotes a 2007 spreadsheet containing macros/VBA code
  - In theory, Excel 2003 can download a free reader that will convert the new formats to the old ones. (Let me know how that works out.)

Excel for the Mac

- Excel for the Mac: a piece of #@&!!. What were they thinking?
  - It doesn't even have VBA.
  - It's almost like Microsoft doesn't like supporting the Mac environment…
  - I run Windows XP on my Mac so I can use Excel 2007.

Three Stages of Spreadsheet Design

- **Think.** Map out your spreadsheet before you begin.
- **Implement.** Code carefully and clearly. Don't be *too* clever.
- **Debug.** Try to *break* the program. Every program ever written has errors. Be the first to find yours.

Spreadsheet Style

- Style matters.
- Design logically, from left to right, from top to bottom.
- In proformas,
  - group like inputs together
  - let columns represent years or other periods
  - don't put new things under old things

Good spreadsheet form: move left to right, and top to bottom.

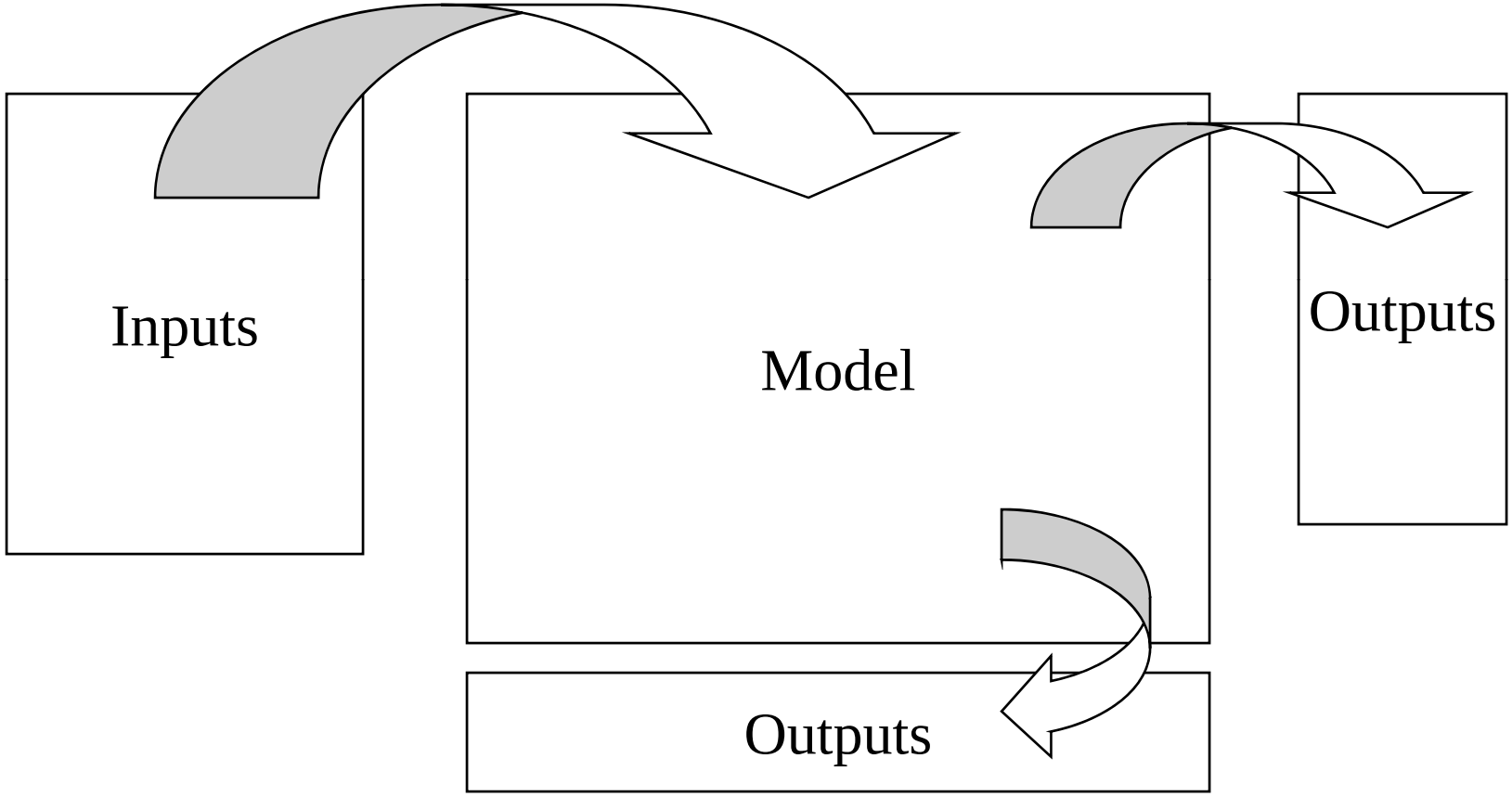


Exhibit 14-2

From Geltner and Miller – can we improve this?

Exhibit 14-2: Example After-Tax Income & Cash Flow Proformas . . .

			Unlevered:	Levered:
Property Purchase Price (Year 0):	$1,000,000			
Depreciable Cost Basis:	$800,000	Before-tax IRR:	6.04%	7.40%
Ordinary Income Tax Rate:	35.00%	After-tax IRR:	4.34%	6.44%
Capital Gains Tax Rate:	15.00%	Ratio AT/BT:	0.719	0.870
Depreciation Recaptu	25.00%			

	Year: 1	2	3	4	5	6	7	8	9	Oper. Yr.10	Reversion Item:	Rever. Yr.10	Total Yr.10
Operating:													
Accrual Items:													
NOI	$60,000	$60,600	$61,206	$61,818	$62,436	$63,061	$63,691	$64,328	$64,971	$65,621	Sale Price	$1,104,622	
- Depr.Exp.	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	- Book Val	$809,091	
- Int.Exp.	$41,250	$41,140	$41,030	$40,920	$40,810	$40,700	$40,590	$40,480	$40,370	$40,260			
=Net Income (BT)	($10,341)	($9,631)	($8,915)	($8,193)	($7,465)	($6,730)	($5,990)	($5,243)	($4,490)	($3,730)	=Book Gain	$295,531	$291,801
- IncTax	($3,619)	($3,371)	($3,120)	($2,867)	($2,613)	($2,356)	($2,096)	($1,835)	($1,571)	($1,305)	- CGT	$73,421	
=Net Income (AT)	($6,722)	($6,260)	($5,795)	($5,325)	($4,852)	($4,375)	($3,893)	($3,408)	($2,918)	($2,424)	=Gain (AT)	$222,111	$219,686
Adjusting Accrual to Reflect Cash Flow:													
- Cap. Imprv. Expdtr.	$0	$0	$50,000	$0	$0	$0	$0	$50,000	$0	$0			
+ Depr.Exp.	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	$29,091	+ Book Val	$809,091	
-DebtAmort	$2,000	$2,000	$2,000	$2,000	$2,000	$2,000	$2,000	$2,000	$2,000	$2,000	-LoanBal	$730,000	
=EATCF	$20,369	$20,831	($28,704)	$21,766	$22,239	$22,716	$23,198	($26,317)	$24,173	$24,667	=EATCF	$301,202	$325,868
+ IncTax	($3,619)	($3,371)	($3,120)	($2,867)	($2,613)	($2,356)	($2,096)	($1,835)	($1,571)	($1,305)	+ CGT	$73,421	
=EBTCF	$16,750	$17,460	($31,824)	$18,898	$19,626	$20,361	$21,101	($28,152)	$22,601	$23,361	=EBTCF	$374,622	$397,983

CASH FLOW COMPONENTS FORMAT

	Year: 1	2	3	4	5	6	7	8	9	Oper. Yr.10	Reversion Item	Rever. Yr.10	Total Yr.10
Operating:													
Accrual Items:													
NOI	$60,000	$60,600	$61,206	$61,818	$62,436	$63,061	$63,691	$64,328	$64,971	$65,621	Sale Price	$1,104,622	
- Cap. Imprv. Expdtr.	$0	$0	$50,000	$0	$0	$0	$0	$50,000	$0	$0			
=PBTCF	$60,000	$60,600	$11,206	$61,818	$62,436	$63,061	$63,691	$14,328	$64,971	$65,621	=PBTCF	$1,104,622	$1,170,243
- Debt Svc	$43,250	$43,140	$43,030	$42,920	$42,810	$42,700	$42,590	$42,480	$42,370	$42,260	- LoanBal	$730,000	
=EBTCF	$16,750	$17,460	($31,824)	$18,898	$19,626	$20,361	$21,101	($28,152)	$22,601	$23,361	=EBTCF	$374,622	$397,983
-taxNOI	$21,000	$21,210	$21,422	$21,636	$21,853	$22,071	$22,292	$22,515	$22,740	$22,967	taxMktGain	$693	$23,661
+ DTS	$10,182	$10,182	$10,182	$10,182	$10,182	$10,182	$10,182	$10,182	$10,182	$10,182	- AccDTS	($72,727)	($62,545)
+ ITS	$14,438	$14,399	$14,361	$14,322	$14,284	$14,245	$14,207	$14,168	$14,130	$14,091			$14,091
=EATCF	$20,369	$20,831	($28,704)	$21,766	$22,239	$22,716	$23,198	($26,317)	$24,173	$24,667	EATCF	$301,202	$325,868

Avoid this.

(from a well-known textbook).

Usually include "period 0," the development or investment period.

Sale is in year 5, so this stuff should be in column for year 5

Generally, sign "cash in" as positive, "cash out" as negative. =SUM positive and negative numbers. Avoid subtraction.

Exhibit 4

EXHIBIT 10–21 Summary Loan Information

	End of Year				
	1	*2*	*3*	*4*	*5*
Payment	$ 689,025	$ 689,025	$ 689,025	$ 689,025	$ 689,025
Mortgage balance	5,851,543	5,742,776	5,622,620	5,489,883	5,343,245
Interest	590,569	580,259	568,869	556,288	542,388
Principal	98,457	108,767	120,156	132,738	146,637

EXHIBIT 10–22 After-Tax Cash Flow from Sale in year 5

Sale price			$9,700,000
Less mortgage balance			5,343,245
Before-tax cash flow (*BTCFs*)			4,356,755
Taxes in year of sale			
Sale price		$9,700,000	
Original cost basis	$8,500,000		
Accumulated depreciation	926,282		
Adjusted basis		7,573,718	
Capital gain		$2,126,282	
Tax on gain at 28%			595,359
After-tax cash flow from sale (*ATCFs*)			$3,761,396

			Stephen Malpezzi	**IN PROGRESS**					
			File Simple Excel Proforma (A Single Case, Ignoring Passive Losses)						
			Preliminary, 03/11/01						
					ANNUAL CASH FLOWS				
MODEL INPUTS			**MORTGAGE**	0	1	2	3	4	5
			Mortgage Payment		(912,840)	(912,840)	(912,840)	(912,840)	(912,840)
	mnemonic		Interest		(664,940)	(641,685)	(616,249)	(588,426)	(557,994)
Size	ZSIZE	100,000	Principal		(247,900)	(271,155)	(296,591)	(324,414)	(354,846)
Gross Rent/SF	ZGR_SF	15	Loan Balance	7,500,000	7,252,100	6,980,945	6,684,353	6,359,940	6,005,094
Gross Potential Income	ZGPI	1,500,000		**Check CUMIPMT:**	(664,940)				
			OPERATIONS	**Check CUMPRINC:**	(247,900)				
Year 1 Vacancy Rate	ZVAC1	30%	Gross Potential Income		1,500,000	1,575,000	1,653,750	1,736,438	1,823,259
Year 2 Vacancy Rate	ZVAC2	10%	Less Vacancy & Collection		(450,000)	(157,500)	(82,688)	(86,822)	(91,163)
Year 3-99 VR	ZVAC3	5%	Effective Gross Income		1,050,000	1,417,500	1,571,063	1,649,616	1,732,096
			Less Operating Expenses		(600,000)	(630,000)	(661,500)	(694,575)	(729,304)
Inflation	ZINFL	5.0%	**Net Operating Income**		**450,000**	**787,500**	**909,563**	**955,041**	**1,002,793**
M&R (% of GPI)	ZMR	40.0%	Less Depreciation		(205,128)	(205,128)	(205,128)	(205,128)	
			Less Interest Payment		(664,940)	(641,685)	(616,249)	(588,426)	
Purchase Price	ZPRICE	-10,000,000	**Taxable Income**		**(420,068)**	**(59,313)**	**88,186**	**161,486**	
Land/Total Cost	ZLSHARE	20.0%	Plus Depreciation		205,128	205,128	205,128	205,128	
IRS Useful Life	ZLIFE	39.0	Less Principal Payment		(247,900)	(271,155)	(296,591)	(324,414)	
			Before Tax Cash Flow		**(462,840)**	**(125,340)**	**(3,277)**	**42,201**	
Income Tax Rate (State & F	ZTR	0.35	Less Taxes (Ignoring Passive Loss Restrictions)		147,024	20,760	(30,865)	(56,520)	
K Gains Rate (State & Fed	ZKTRATE	0.20	**After Tax Cash Flow, Operations**		**(315,816)**	**(104,580)**	**(34,142)**	**(14,319)**	
K Gains Rate Depreciation	ZKTRDEP	0.25							
			REVERSION CASH FLOW						
LTV	ZLTV	75.00%	*Capital Gain Part I, Neglecting Accumulated Depreciation*						
Interest	ZINT	9.00%	Gross Sales Price		**IN PROGRESS**			10,027,927	
Loan Term	ZTERM	15.0	Less Selling Cost					(501,396)	
			Less Purchase Price					(10,000,000)	
			Capital Gain, Part I					**(473,470)**	
Final Cap Rate	ZCAP	10.0%							
Sale Commission	ZCOM	5.0%	*Capital Gain Part II, Adjustment for Accumulated Depreciation*						
			Accumulated Depreciation					820,513	
Required Return	ZYIELD	15.0%							
			Tax on Part I of Capital Gains					94,694	
			Tax on Part II, Accumulated Depreciation					(205,128)	
			Sales Price					10,027,927	
			Less Selling Cost					(501,396)	
			Less Capital Gains Tax					(110,434)	
			Less Outstanding Mortgage					(6,359,940)	
			Net Sales Price (After Tax Equity Reversion)					**3,056,156**	
			TOTAL CASH FLOWS						
			TOTAL ATCF	**(2,500,000)**	**(315,816)**	**(104,580)**	**(34,142)**	**3,041,837**	
			NPV	**(988,669)**					
			IRR	**0.77%**					
			Debt Service Ratios		0.49	0.86	1.00	1.05	

Elements of Spreadsheet Style

- Choose a suitable design and hold to it.
- Use lots of comments and labels. Make your spreadsheets self-documenting.
- Use named ranges for inputs. Make the names mnemonic.
- Be very careful about signs.
  - ↑Cash in = +
  - ↑Cash out = –
- Consider replacing complicated formulae with a macro.

More Elements of Spreadsheet Style

- Never use more than one font.
- Serif fonts (like Times Roman) are more formal, and easier to read in large blocks of text. (The serifs aid recognition).
- Sans serif fonts (like Arial) are easier to read from a distance (e.g. overhead presentations in a large room).
- Use color, italics, etc. sparingly.
  - If for some reason you violate these rules, ***never*** use more than two fonts.

A Few Cool Things Most Spreadsheet Users Neglect

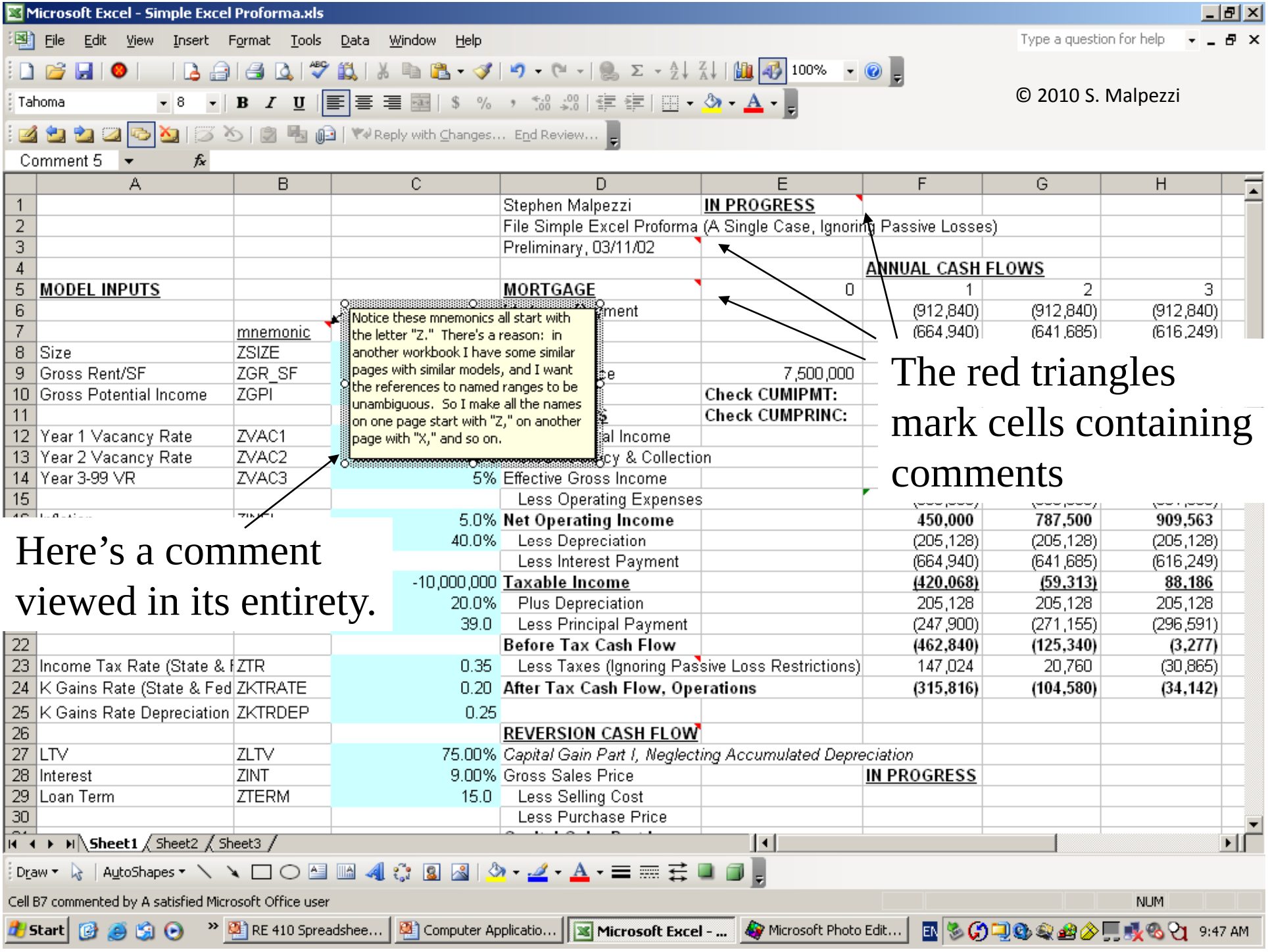
- Comments
- Named Ranges
- The Auditing Tool
- Scenario Manager

User Friendly, Professional Models

- *Subtly* highlight input areas, and key output areas
- Consider protecting cells containing formulas.
  - *Password* protected cells if you don't want anyone messing with them.
- Consider using the validation tools to limit data entry ranges to valid values.
- Use lots of comments. Make your spreadsheet self-documenting.

Comments

- Professional spreadsheets are clear and well-documented.
- Sometimes you will document with text contained in cells.
- For major projects, a written user's guide may be required.
- Another useful tool are comments within cells.
- 2003: Select the cell, then "INSERT, COMMENT"
  - (To control default view – red triangles, full comment, or nothing – "TOOLS, OPTIONS, VIEW")
- 2007: Select the cell, then "REVIEW tab of the Ribbon, NEW COMMENT" in the Comments Group.



The red triangles mark cells containing comments

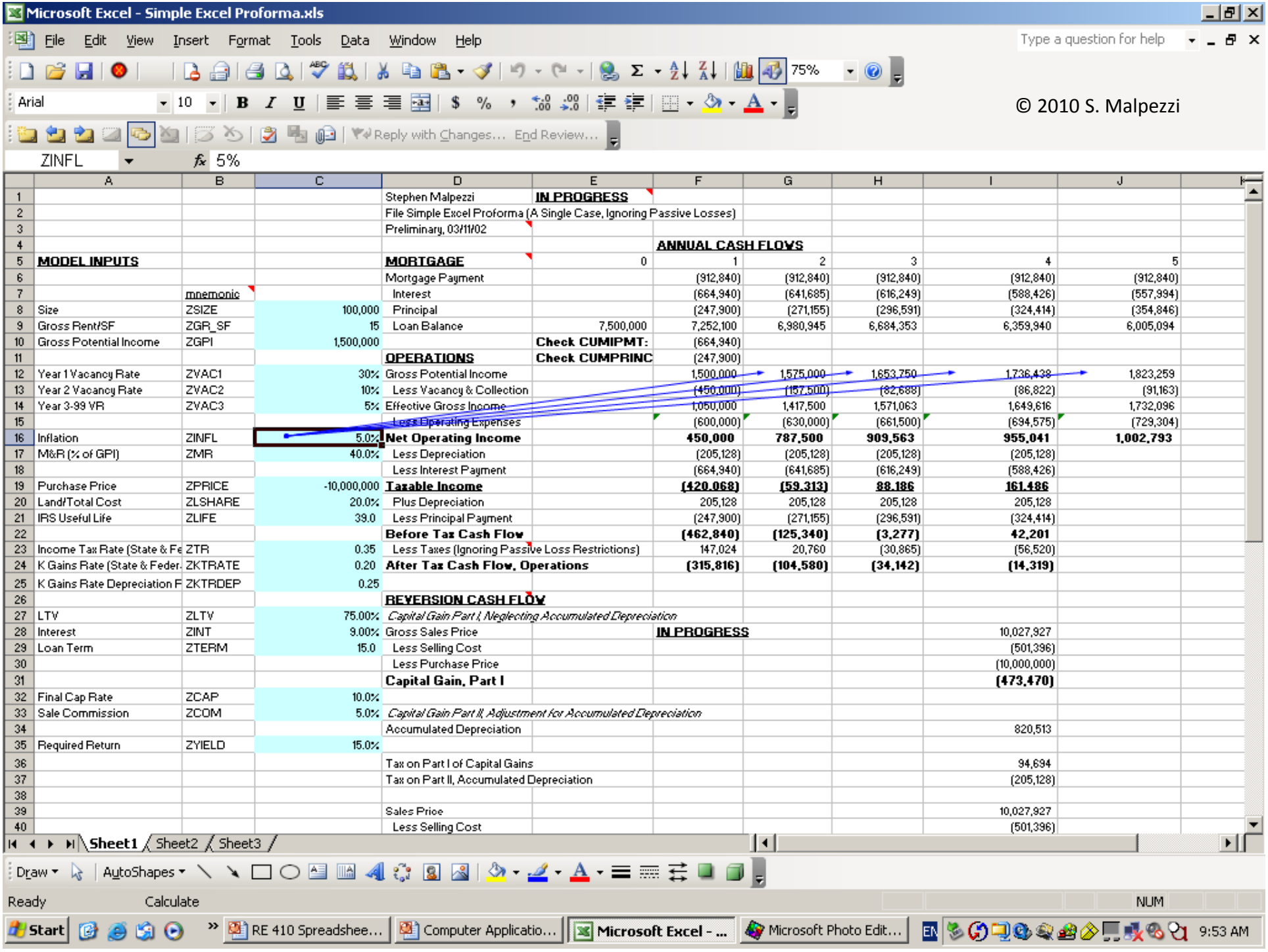
Here's a comment viewed in its entirety.

Be cool: use the auditing tools

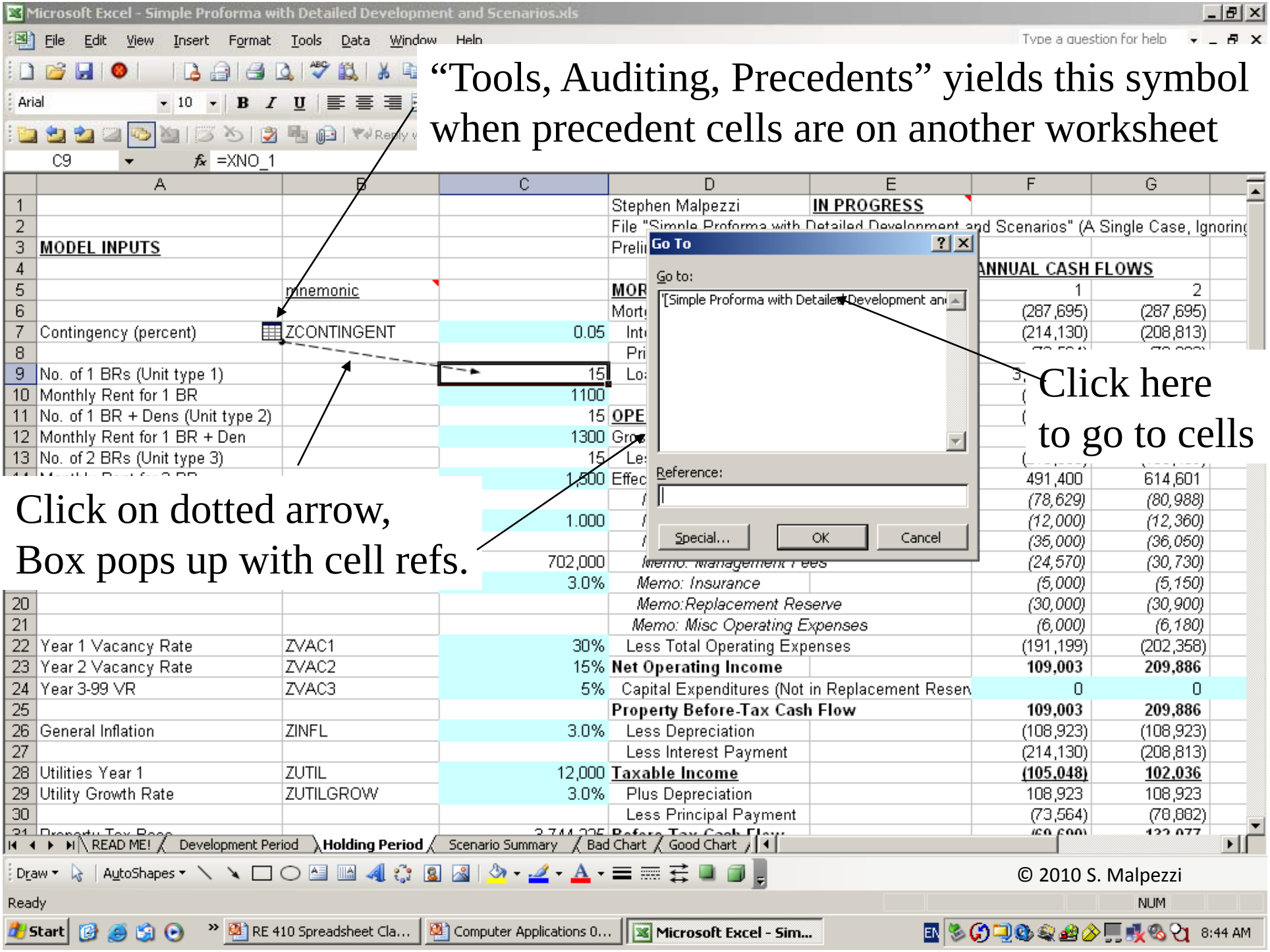
- *Trace Precedents* shows which cells contain inputs to the selected cell.
- *Trace Dependents* shows what cells are based on the data in the current cell.
- 2003: TOOLS Menu
- 2007: FORMULAS tab of the Ribbon

Auditing Tools

- Tools, Auditing, Trace Precedents: find cells that are the building blocks of the current cell.
- Tools, Auditing, Trace Dependents: find cells that depend on the current cell.
  - Unfortunately, auditing only points to cells within a single worksheet.
- Tools, Formula Auditing: show formulae on spreadsheet instead of results (toggle back and forth with Ctrl+ ‘



	A	B	C	D	E	F	G	H	I	J
1				Stephen Malpezzi	**IN PROGRESS**					
2				File Simple Excel Proforma (A Single Case, Ignoring Passive Losses)						
3				Preliminary, 03/11/02						
4						**ANNUAL CASH FLOWS**				
5	**MODEL INPUTS**			**MORTGAGE**	0	1	2	3	4	5
6				Mortgage Payment		(912,840)	(912,840)	(912,840)	(912,840)	(912,840)
7		mnemonic		Interest		(664,940)	(641,685)	(616,249)	(588,426)	(557,994)
8	Size	ZSIZE	100,000	Principal		(247,900)	(271,155)	(296,591)	(324,414)	(354,846)
9	Gross Rent/SF	ZGR_SF	15	Loan Balance	7,500,000	7,252,100	6,980,945	6,684,353	6,359,940	6,005,094
10	Gross Potential Income	ZGPI	1,500,000		**Check CUMIPMT:**	(664,940)				
11				**OPERATIONS**	**Check CUMPRINC**	(247,900)				
12	Year 1 Vacancy Rate	ZVAC1	30%	Gross Potential Income		1,500,000	1,575,000	1,653,750	1,736,438	1,823,259
13	Year 2 Vacancy Rate	ZVAC2	10%	Less Vacancy & Collection		(450,000)	(157,500)	(82,688)	(86,822)	(91,163)
14	Year 3-99 VR	ZVAC3	5%	Effective Gross Income		1,050,000	1,417,500	1,571,063	1,649,616	1,732,096
15				Less Operating Expenses		(600,000)	(630,000)	(661,500)	(694,575)	(729,304)
16	Inflation	ZINFL	5.0%	**Net Operating Income**		**450,000**	**787,500**	**909,563**	**955,041**	**1,002,793**
17	M&R (% of GPI)	ZMR	40.0%	Less Depreciation		(205,128)	(205,128)	(205,128)	(205,128)	
18				Less Interest Payment		(664,940)	(641,685)	(616,249)	(588,426)	
19	Purchase Price	ZPRICE	-10,000,000	**Taxable Income**		**(420,068)**	**(59,313)**	**88,186**	**161,486**	
20	Land/Total Cost	ZLSHARE	20.0%	Plus Depreciation		205,128	205,128	205,128	205,128	
21	IRS Useful Life	ZLIFE	39.0	Less Principal Payment		(247,900)	(271,155)	(296,591)	(324,414)	
22				**Before Tax Cash Flow**		**(462,840)**	**(125,340)**	**(3,277)**	**42,201**	
23	Income Tax Rate (State & Fe	ZTR	0.35	Less Taxes (Ignoring Passive Loss Restrictions)		147,024	20,760	(30,865)	(56,520)	
24	K Gains Rate (State & Feder	ZKTRATE	0.20	**After Tax Cash Flow, Operations**		**(315,816)**	**(104,580)**	**(34,142)**	**(14,319)**	
25	K Gains Rate Depreciation F	ZKTRDEP	0.25							
26				**REVERSION CASH FLOW**						
27	LTV	ZLTV	75.00%	*Capital Gain Part I, Neglecting Accumulated Depreciation*						
28	Interest	ZINT	9.00%	Gross Sales Price		**IN PROGRESS**			10,027,927	
29	Loan Term	ZTERM	15.0	Less Selling Cost					(501,396)	
30				Less Purchase Price					(10,000,000)	
31				**Capital Gain, Part I**					**(473,470)**	
32	Final Cap Rate	ZCAP	10.0%							
33	Sale Commission	ZCOM	5.0%	*Capital Gain Part II, Adjustment for Accumulated Depreciation*						
34				Accumulated Depreciation					820,513	
35	Required Return	ZYIELD	15.0%							
36				Tax on Part I of Capital Gains					94,694	
37				Tax on Part II, Accumulated Depreciation					(205,128)	
38										
39				Sales Price					10,027,927	
40				Less Selling Cost					(501,396)	

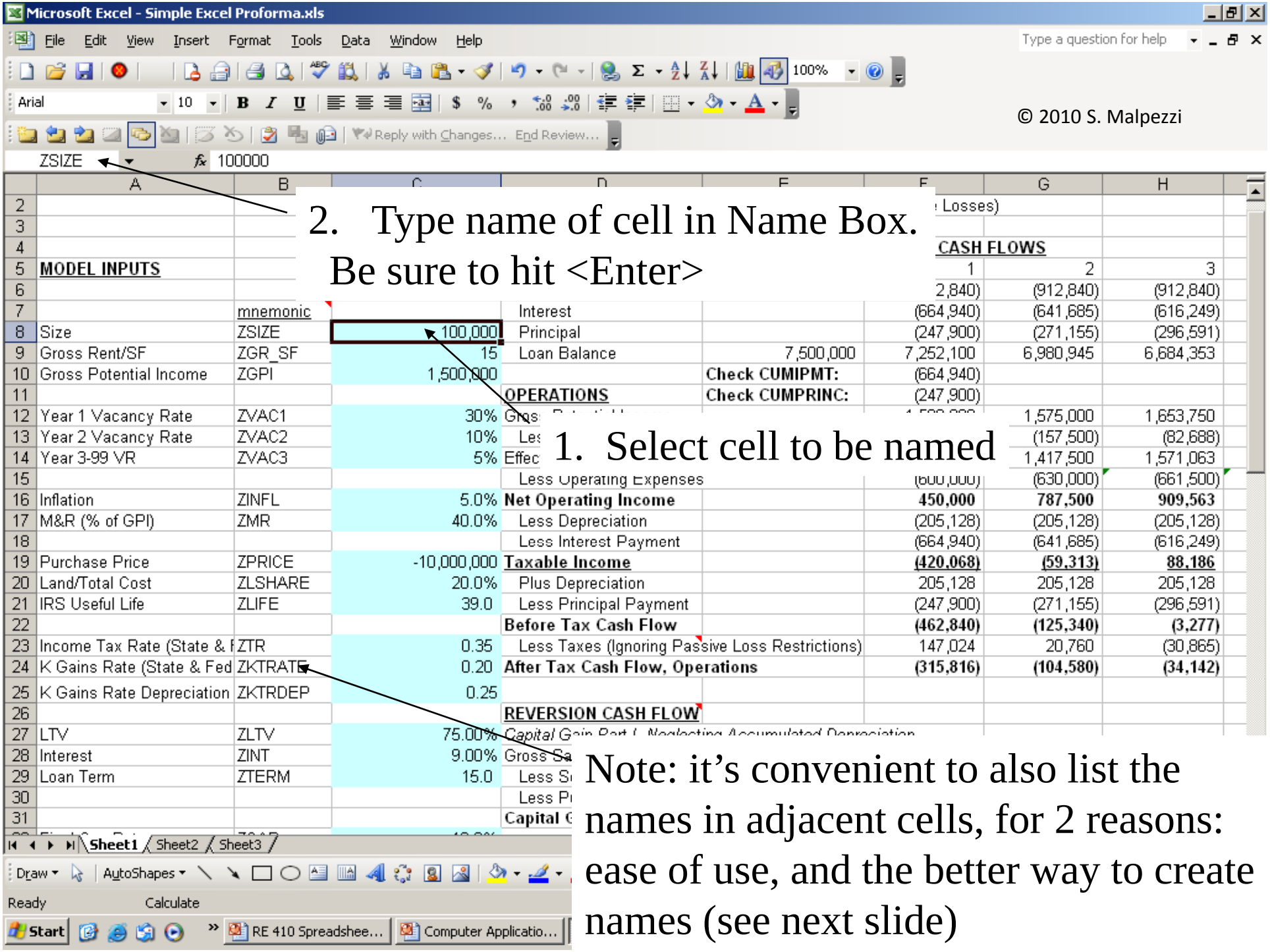

"Tools, Auditing, Precedents" yields this symbol when precedent cells are on another worksheet
Click here to go to cells
Click on dotted arrow, Box pops up with cell refs.
Microsoft Excel - Simple Proforma with Detailed Development and Scenarios.xls
Go To
Go to:
Reference:
Special...
OK
Cancel
READ ME!
Development Period
Holding Period
Scenario Summary
Bad Chart
Good Chart
© 2010 S. Malpezzi

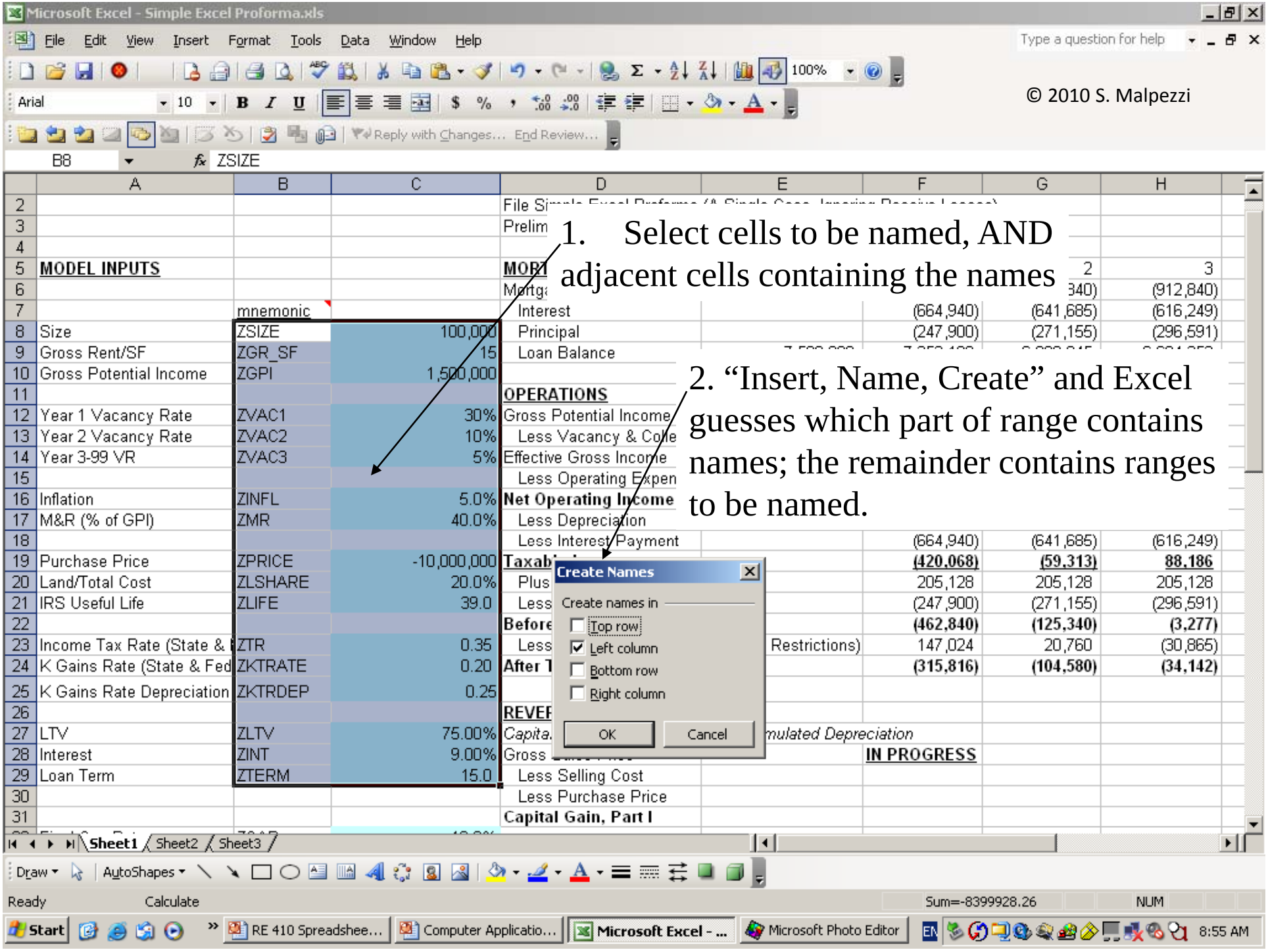
Named Ranges

- The most common way to refer to cells is to use the labels of columns and rows on a worksheet.
  - For example, Sheet1!C27
- Or you can create descriptive names to represent cells, ranges of cells, formulas, or constant values. Often, we use *mnemonic* names.
  - Example: ZLTV

A few ways to create named ranges

- Select the cell, or cells, to be named. Then type the name in the NAME BOX that appears in Excel just above the top left.
- Select the cells to be named, ***as well as*** adjacent cells (usually above, or to the left) of those cells that contain the text of the names. Then go to the drop down menus and INSERT, NAME, CREATE, OK (2003).
  - 2007: FORMULAS tab of the Ribbon, "DEFINED NAMES" section.


Microsoft Excel - Simple Excel Proforma.xls
ZSIZE
100000
2. Type name of cell in Name Box.
Be sure to hit <Enter>
1. Select cell to be named
Note: it's convenient to also list the names in adjacent cells, for 2 reasons: ease of use, and the better way to create names (see next slide)
© 2010 S. Malpezzi


Microsoft Excel - Simple Excel Proforma.xls
File Edit View Insert Format Tools Data Window Help
Type a question for help
100%
Arial 10
Reply with Changes... End Review...
B8 ZSIZE
© 2010 S. Malpezzi
1. Select cells to be named, AND adjacent cells containing the names
2. "Insert, Name, Create" and Excel guesses which part of range contains names; the remainder contains ranges to be named.
Create Names
Create names in
Top row
Left column
Bottom row
Right column
OK
Cancel
MODEL INPUTS
mnemonic
Size ZSIZE 100,000
Gross Rent/SF ZGR_SF 15
Gross Potential Income ZGPI 1,500,000
Year 1 Vacancy Rate ZVAC1 30%
Year 2 Vacancy Rate ZVAC2 10%
Year 3-99 VR ZVAC3 5%
Inflation ZINFL 5.0%
M&R (% of GPI) ZMR 40.0%
Purchase Price ZPRICE -10,000,000
Land/Total Cost ZLSHARE 20.0%
IRS Useful Life ZLIFE 39.0
Income Tax Rate (State & ZTR 0.35
K Gains Rate (State & Fed ZKTRATE 0.20
K Gains Rate Depreciation ZKTRDEP 0.25
LTV ZLTV 75.00%
Interest ZINT 9.00%
Loan Term ZTERM 15.0
OPERATIONS
IN PROGRESS
Sheet1 Sheet2 Sheet3
Draw AutoShapes
Ready Calculate
Sum=-8399928.26 NUM
Start
RE 410 Spreadshee...
Computer Applicatio...
Microsoft Excel - ...
Microsoft Photo Editor
8:55 AM

Advantages of named ranges

- It's easier to remember ZLTV than Sheet1!C27.
- Unlike row and column labels, named ranges remain unchanged as rows and columns are inserted or deleted, or otherwise moved around.
- When we use VBA (later), named ranges are nearly essential.
- Note: a name can refer to a single cell, a range of cells. Ranges can even be non-contiguous, and extend over more than one worksheet.
- Easier to read formulae.

My Rules For Real Estate Investing

By Martha Stewart

Garbage in, Garbage out.

- Choosing the right decision rule(s) is important.
  - e.g. IRR, NPV, are better primary rules than "cash-on-cash returns", though the latter is of some interest.
- But data, information and judgment are even more important.
  - Good data and a rule of thumb can outperform lousy data and the best calculation technique.
  - Holistic: if primary (IRR, NPV) and secondary (DCR, cash on cash) "disagree," ask why?

Everything you know is wrong!

- No matter how good your estimates and forecasts – of future NOI, of going-out cap rates, vacancy rates, etc. – we know one thing *for certain*: your forecasts will be *wrong*.
- Forget "the" rate of return or "the" NPV. Think of ranges, probabilities.
- Many ways to take this on, Scenario Manager is a useful start.
  - To get started in 2003: TOOLS, SCENARIOS, ADD…
  - In 2007: DATA tab of the Ribbon, DATA TOOLS section, "WHAT-IF ANALYSIS"

Scenario Manager

- At a minimum, I define three scenarios: my "likely" scenario, "optimistic case," and "worst case."
  - Likely: my *best* estimates and forecasts
  - Optimistic: a good day at the office, but still reasonable.
  - Worst Case: vacancies and interest rates rising, rents falling, a good day to teach rather than invest.
- To get started: "Tools, Scenario, Add" then follow through menus. Obviously, you'll want to create at least two scenarios.
- *Trap: IF Calculation is set to Manual, Scenario Manager doesn't work. In the Tools menu, set Calculation to Automatic.*

Another scenario (or set of scenarios) you should include.

- What assumptions drive you to the threshold of default? (EG to a debt coverage ratio less than one? Where the value of the building falls below the value of the mortgage?)
  - Can undertake trial and error.
  - Can use SOLVER to find assumptions that drive you to the threshold.
- Note: If you have (say) two key inputs (rent, and stabilized vacancy rate) and only one criterion for default (DCR), there are many possible solutions.
  - Simple solution: you can either fix rents and change vacancy, or the reverse.
  - More sophisticated: you can specify a relationship between rents and vacancies within Solver.

Scenario Manager Output

Named Ranges would be good here!!!

Scenario Summary	Current Values:	High	Medium	Low
Changing Cells:				
ZGR_SF	15	16	15	13
ZVAC1	30%	30%	25%	40%
ZVAC2	10%	10%	10%	20%
ZVAC3	5%	5%	5%	10%
Result Cells:				
I28	10,027,927	10,696,455	10,027,927	7,900,791
E48	(988,669)	(655,769)	(951,807)	(2,126,186)
E49	0.77%	6.07%	1.21%	-24.10%
F50	0.49	0.53	0.58	0.28

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

Alternatives to Scenario Manager

- Data Tables
  - Also in “What-If” section. Figure out on your own.

Common functions you should know cold.

- SUM, AVERAGE, MEDIAN, PERCENTILE
- PV, NPV, IRR
- PMT, FV
- LN, EXP
- IF
- How to use SOLVER (not a function, strictly, but extremely useful).

Some tips on familiar functions (I)

- Got data? Great! How to summarize it? Measures of central tendency are the first step!
- Everyone is familiar with average. Consider using medians and other "order statistics"
- **AVERAGE**(**value1**,value2,...)
- Medians are better measures of central tendency for variables that are very skewed – like real estate values, incomes…
- **MEDIAN**(**number1**,number2,...)
- A more versatile function computes medians, quartiles, deciles, or any percentile-based statistic:
- **PERCENTILE**(**array**,**k**)
  - To compute a median with PERCENTILE, set k=.5

Some tips on familiar functions (II)

- Remember the differences between PV and NPV:
  - PV is the present value of a repeated payment of constant size; NPV is the present value of a stream of varying payments.

  PV(**rate**,**nper**,**pmt**,fv,type)

  NPV(**rate**,**value1**,value2, ...)
- Remember that NPV starts discounting immediately. If your investment is made today (undiscounted), compare:

  =NPV(–300,100,100,100) is incorrect

  =–300+NPV(100,100,100) is correct

IF Function

- Returns one value if a condition you specify evaluates to TRUE and another value if it evaluates to FALSE.
- Use IF to conduct conditional tests on values and formulas.
- **IF**(**logical_test**,**value_if_true**,value_if_false)

Some very handy, but under-appreciated, functions

- RAND (random number generator)
- MIRR (modified IRR, accounting for finance and reinvestment interest)
- XIRR, XNPV (IRR and NPV for irregular cash flows)
- ISERROR (returns an Excel error value; use with IF, as in:

  =IF(ISERROR(B12),"We Are Doomed","Life is Good")

What if you can't find some advanced functions? "Add-ins."

- Advanced functions are part of the ANALYSIS TOOLPACK in Excel. They are "Add-Ins."
  - Add-ins are actually macros and functions written in VBA, and "compiled" (you can use them but can't readily access the code).
  - Some Add-ins are available commercially (e.g. @RISK or Crystal Ball or S-Plus).
  - Once you know VBA you can write your own
  - Some Add-Ins are provided by Microsoft but have to be activated.

Activating “Add-ins.”

- In 2003, TOOLS, ADD-INS, then check off the Add-ins you desire.
- 2007, HOME tab (top left corner), EXCEL OPTIONS, ADD-INS; then go to to the MANAGE ADD-INS button at the bottom. Then check off the Add-ins you need.
- Of readily available Microsoft Add-ins I recommend you check off ANALYSIS TOOLPACK, ANALYSIS TOOLPACK-VBA, and SOLVER.

Other Useful Functions: "IS" functions.

- ISBLANK(cell ref.): is a cell blank?
- ISERROR(cell ref.): does a cell return an error?
- ISNUMBER(cell ref): does a cell contain a number?
- ISTEXT(cell ref): does a cell contain text?
  - And five other variations on the theme (see online help).

A common problem in proformas (and other models)

- Sometimes we want to enter a value of a variable that will remain constant for all years or periods.
  - EG inflation rates, mortgage rates, vacancy rates…
- But sometimes we want to override the default value.
  - What if inflation takes off? What if we have a variable rate mortgage? What if vacancy rates are falling as our property leases up, or rising as the market tanks?
- Two-part solution:
  - Set up a data entry form that allows you to enter an initial rate, that you can choose to override for one or more periods.
  - When using the variable, combine IF and ISBLANK functions to use entered values, or default if no value entered.

Example: two ways to enter an inflation adjustment

- (1) Default inflation rate for a given line item (Column I of spreadsheet fragment in next slide)
- (2) Inflation rate for a given year for a given line item
  - If (2) is entered, that entry overrides the default in (1)
- This example: inflation adjustments for various rents or expenses. But this structure can be used for any variable that might be assumed constant for all periods, or alternatively might vary year by year (e.g. vacancy rate, mortgage rate…)

Setup for inflation adjustment: input data

When cell is blank, will default to value in Col. I of this row:

	A	C	E	G	H	I	J	K	L	M
1	PRO FORMA OPERATING ASSUMPTIONS									
2	APARTMENT INVESTMENT									
3	Merz Apartments									
4	555 Main Street									
6	RENTAL INCOME ASSUMPTIONS									
7				Monthly Rent		GROWTH RATE ASSUMPTIONS				
8		Unit Type	# Units	Year 1		Stabilized	Year 2	Year 3	Year 4	Year 5
9		Efficiency	15	$ 700		4.0%				
10		1BR1BA	25	$ 750		4.0%	6.0%	6.0%		
11		2BR1BA	25	$ 800		4.0%				
12		2BR2BA	15	$ 825		4.0%				

When NOT blank, value in this cell overrides value in Col. I

Now use data from previous slide in formulae

	A	B	C	D	E	F	G
1							Mertz Apartments Operating
2							555 Main Street
3							
4				Year 1	Year 2	Year 3	Year 4
5	GROSS POTENTIAL INCOME						
6	Rental Income						
7		Efficiency		$ 126,000	$ 131,040	$ 136,282	$ 141,733
8		1BR1BA		225,000	238,500	252,810	262,922
9		2BR1BA		240,000	249,600	259,584	269,967
10		2BR2BA		148,500	154,440	160,618	167,042

=E9*(1+(IF(ISBLANK('Inputs-Oper'!K11),'Inputs-Oper'!$I11,'Inputs-Oper'!K11)))

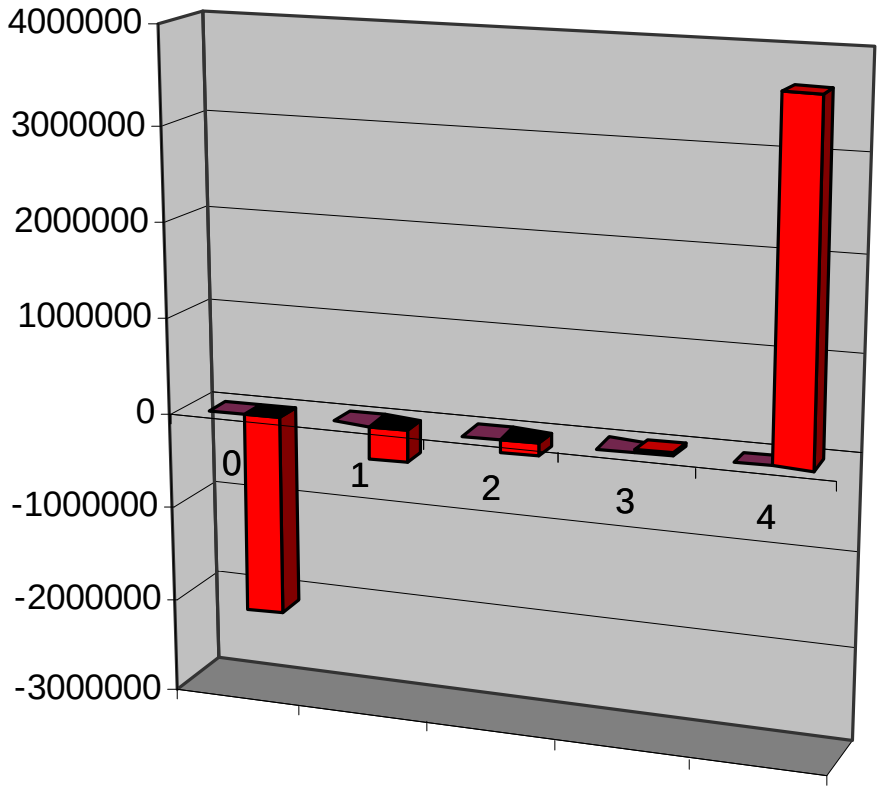
Another Design Decision: *when to start another worksheet?*

- Do we stick with a one-sheet model, or put inputs on one page, the cash flow model on a second page, outputs on a third? Depends.
- Example: what if we develop a building, instead of purchasing a completed building?

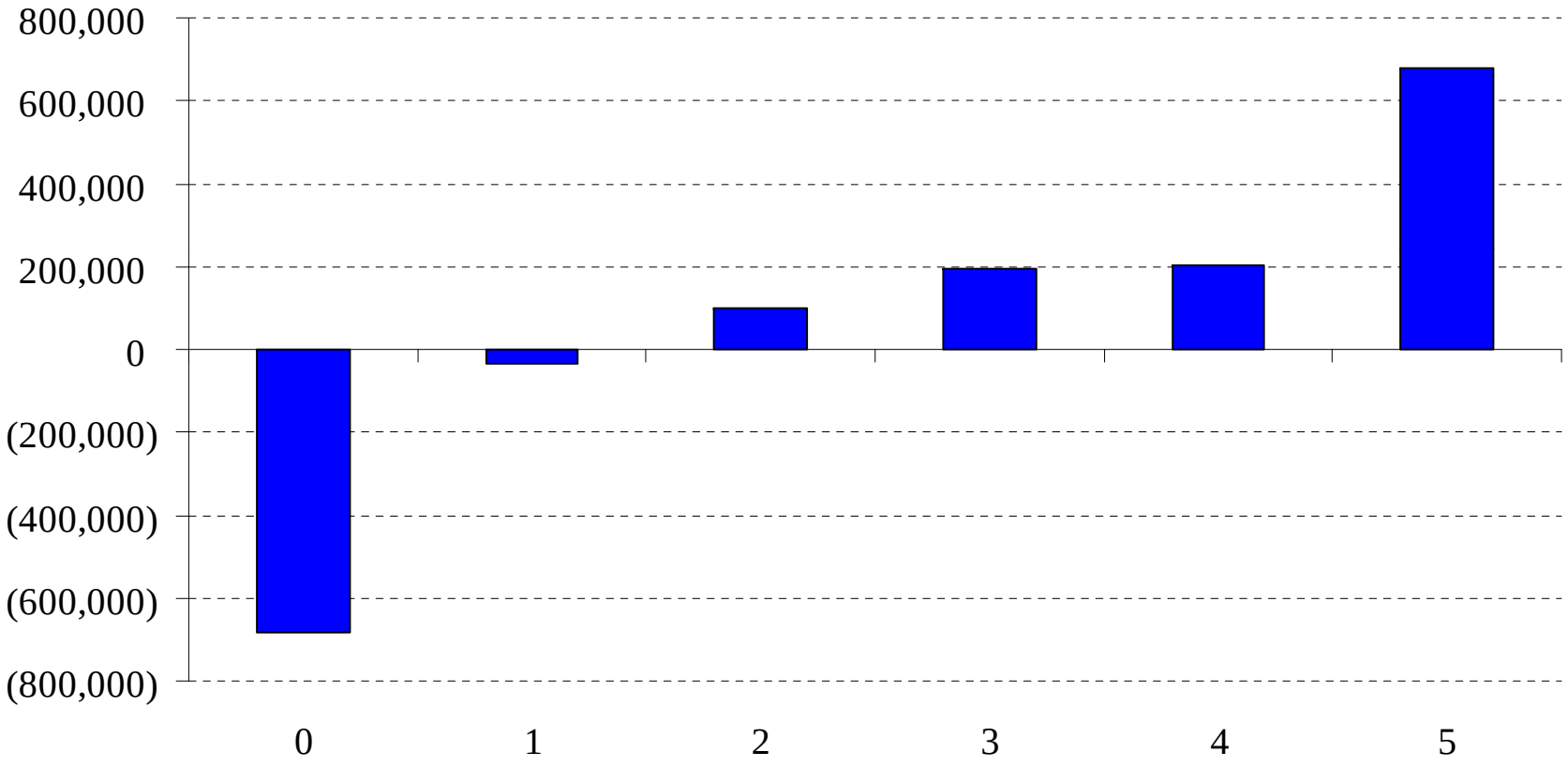
Charting Style

- Keep it simple: avoid 3-D like the plague
- Keep backgrounds uncluttered, and white
  - How will this look when it's xeroxed?
- Use simple, widely available fonts (Arial or Times Roman)
- Limit use of "boxes"
- Simple orientation is usually best

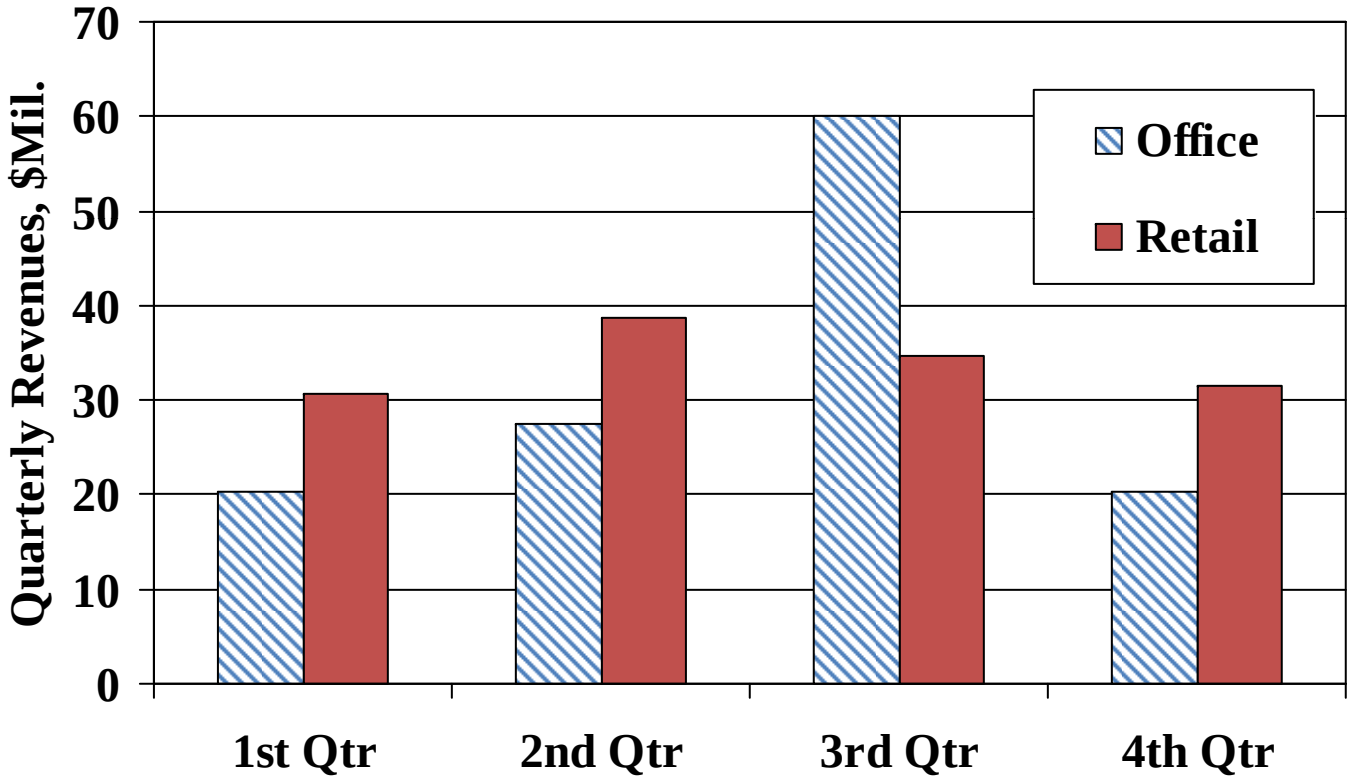
After Tax Cash Flows



After Tax Cash Flows



Revenues by Property Type, Firefly Realty Co. LLP



Charting Tips

- Be sure you understand the difference between line charts and x-y charts.
- Make fonts big enough to read.
  - Overheads need larger fonts than printed reports!

More Charting Tips

- Excel/PowerPoint's charting engine contains many chart types that should never be used
  - Radar charts, cone charts, etc.
- Excel/PowerPoint's charting engine is missing many useful chart types
  - Boxplots, side-by-side graphs, etc.
- Example: use a side-by-side graph if comparisons between individual pairs of values are most important.

Recommendation

- Excel/PowerPoint chart engine is simply not sufficient for serious business users.
- If you are charting data, consider one or more of the following separate products, Excel add-ins, or their equivalents
  - S-Plus (statistics package that has good graphics add-in for Excel)
  - SPSS
  - SigmaPlot (good analytical charts)

Still More Charting Tips

- Excel/PowerPoint's charting engine is particularly poor on "analytic"graphing capabilities
  - Example: there are no true data labels available for x-y plots in Excel
    - What MS calls a data label is not a true data label. Excel allows you to show the value of the y axis variable on the chart; occasionally useful, but a true data label permits use of a third variable.
  - Example: you want to plot house price by MSA population; label points with three letter code for name of MSA. Can't do easily in Excel, although you can easily do it in every other spreadsheet on the market.
  - Solution: use Quattro Pro, or Lotus; or use a macro or an add-in.

Some features of Excel you should learn.

- **VBA: The macro language for Excel**
- Database (list) processing
  - Sorting
  - Filtering (querying) the data
  - VLOOKUP function (turns Excel's flat file databases into the poor man's relational database)
- Data tables (what if)
- Scenarios (what if)
- Pivot tables (crosstabs)
- Solver (optimization)
- Array processing (matrix algebra)

Introduction to Visual Basic for Applications (VBA)

Objectives

- Introduction to VBA
  - The basic idea
  - The environment (the VBA editor)
  - The macro recorder, vs. writing code
  - Key concepts
  - Three kinds of subroutines
- Start an example: a proforma model for multiple cases

VBA Readings

- S. Malpezzi, “Introduction to Macros and Visual Basic in Excel”
- S. Malpezzi, “Deconstructing a Visual Basic Program”
- S. Malpezzi, “Towards a Deeper Understanding of VBA”
- Microsoft, “Making Decisions”

Security considerations and VBA

- Within Excel Options, go to the “Trust Center” and within “Macro Settings,” choose “Disable All Macros with Notification.
- In Excel 2003, we save all files – with or without macros – with extension .xls
- In Excel 2007, if we have a macro, we can’t save with the default .xlsx extension; we must save it with extension .xlsm
- Finally, once you’ve got these right, you may have to save and exit, reopen spreadsheet, and respond to the “Security Warning Macros Have Been Disabled” with “Enable This Content”

Macros: motivation

- How can we automate our work?
- Examples:
  - We design a proforma for a representative investment. What if we have a database with 500 investments?
  - We want to write our own function that calculates IRR for a wide range of possible values

Two ways to create macros in Excel

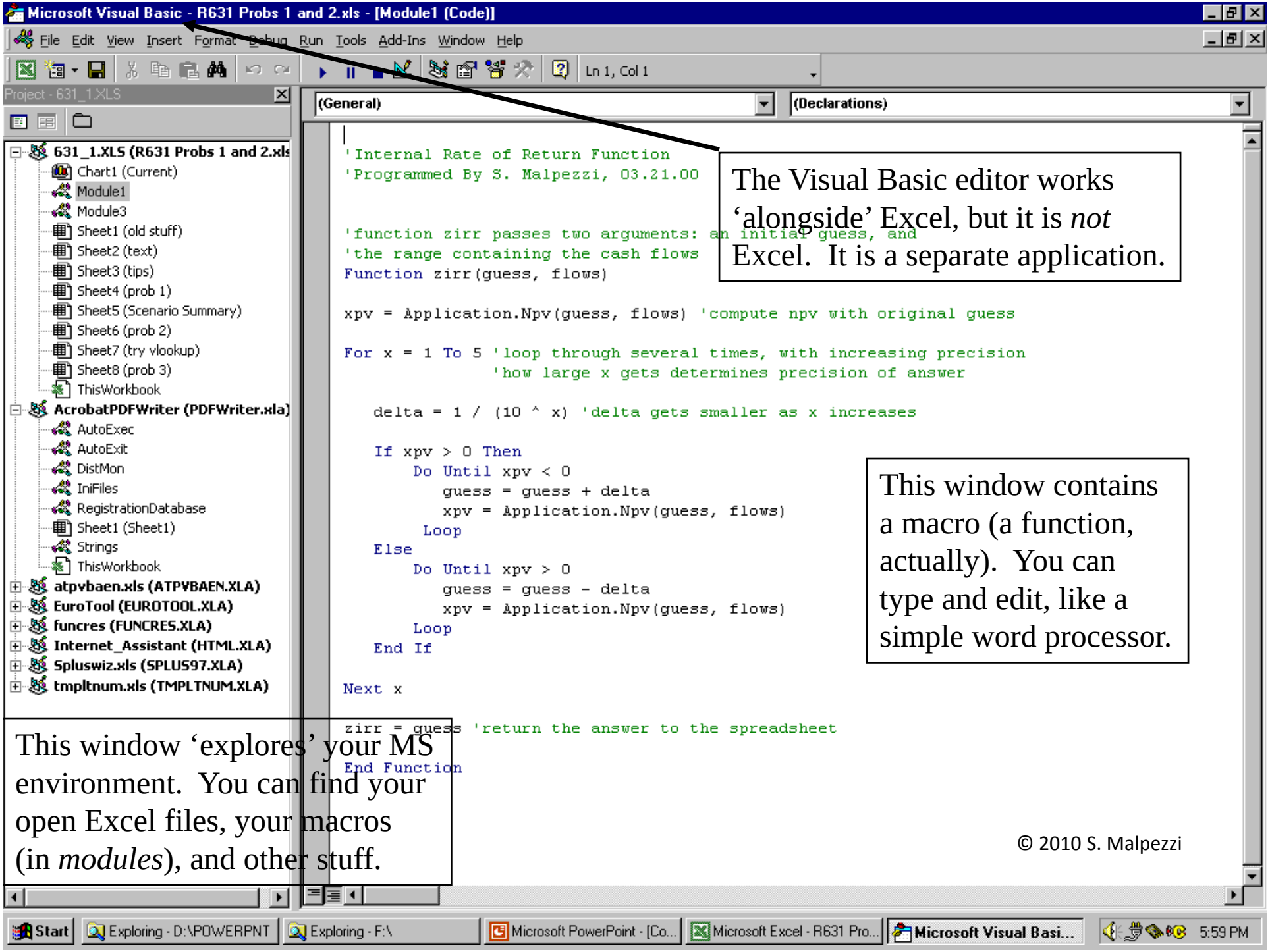
- (1) Use the macro recorder
  - Excel records VBA code that reflects what you do until you turn the recorder off.
- (2) Write code directly in the VBA editor
- Usually, we do a bit of both
  - The recorder often does the unexpected, or the redundant; run the recorder, then edit the results.
  - When writing code, and you don't know what to do, try the recorder to get some ideas.

The VBA Editor

- Between Excel 5.0 and Excel 97, the environment for writing VBA code changed a lot.
  - The 97/2000/2002 VBA editor is much more powerful than “module sheets” in 5.0.
  - It is also more complicated.

Getting to Macros and the VBA Editor

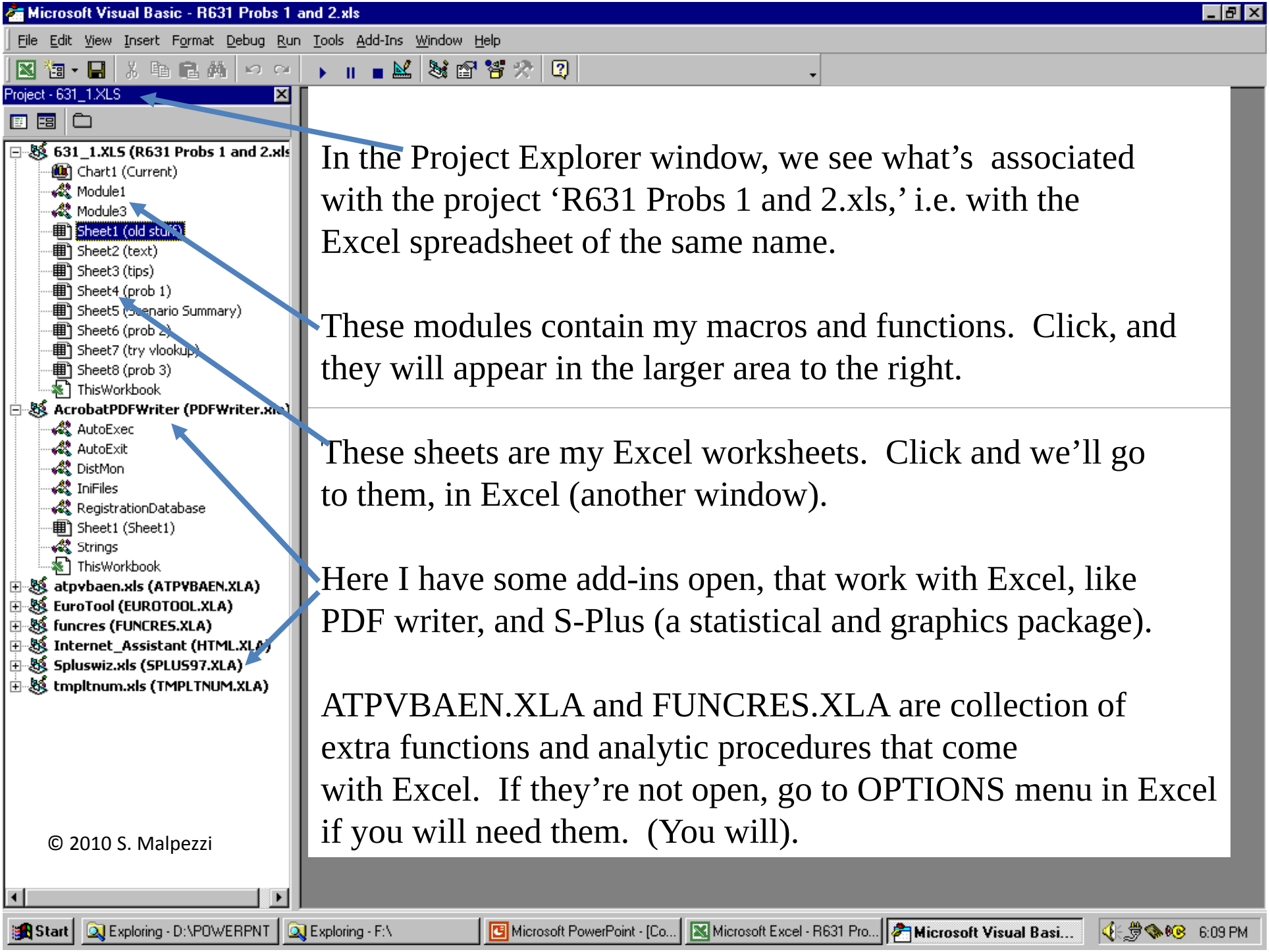
- In 2007: DEVELOPER tab of the Ribbon, CODE Section.
- In 2003: TOOLS, MACRO



The Visual Basic editor works 'alongside' Excel, but it is *not* Excel. It is a separate application.

This window contains a macro (a function, actually). You can type and edit, like a simple word processor.

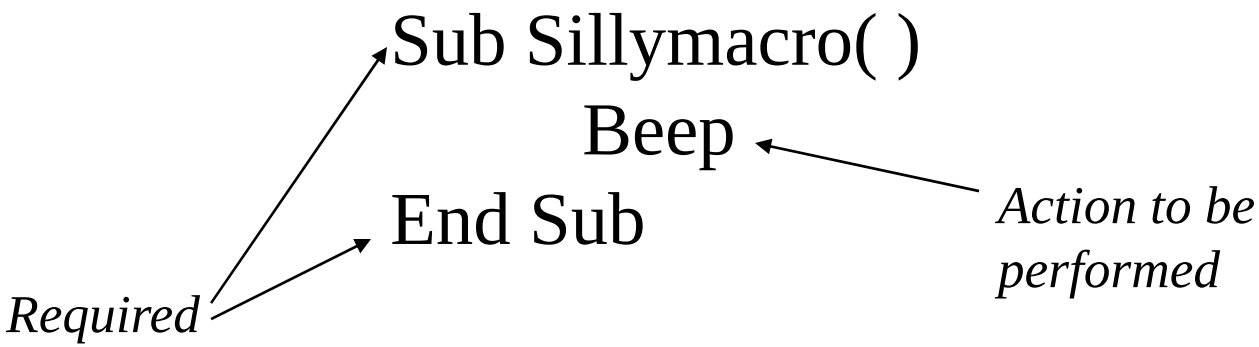
This window 'explores' your MS environment. You can find your open Excel files, your macros (in *modules*), and other stuff.


Microsoft Visual Basic - R631 Probs 1 and 2.xls
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Project - 631_1.XLS
631_1.XLS (R631 Probs 1 and 2.xls
Chart1 (Current)
Module1
Module3
Sheet1 (old stu
Sheet2 (text)
Sheet3 (tips)
Sheet4 (prob 1)
Sheet5 (Scenario Summary)
Sheet6 (prob 2)
Sheet7 (try vlookup)
Sheet8 (prob 3)
ThisWorkbook
AcrobatPDFWriter (PDFWriter.xl
AutoExec
AutoExit
DistMon
IniFiles
RegistrationDatabase
Sheet1 (Sheet1)
Strings
ThisWorkbook
atpvbaen.xls (ATPVBAEN.XLA)
EuroTool (EUROTOOL.XLA)
funcres (FUNCRES.XLA)
Internet_Assistant (HTML.XLA)
Spluswiz.xls (SPLUS97.XLA)
tmpltnum.xls (TMPLTNUM.XLA)
In the Project Explorer window, we see what's associated with the project 'R631 Probs 1 and 2.xls,' i.e. with the Excel spreadsheet of the same name.
These modules contain my macros and functions. Click, and they will appear in the larger area to the right.
These sheets are my Excel worksheets. Click and we'll go to them, in Excel (another window).
Here I have some add-ins open, that work with Excel, like PDF writer, and S-Plus (a statistical and graphics package).
ATPVBAEN.XLA and FUNCRES.XLA are collection of extra functions and analytic procedures that come with Excel. If they're not open, go to OPTIONS menu in Excel if you will need them. (You will).
© 2010 S. Malpezzi
Start
Exploring - D:\POWERPNT
Exploring - F:\
Microsoft PowerPoint - [Co...
Microsoft Excel - R631 Pro...
Microsoft Visual Basi...
6:09 PM

When you open VBA editor for the first time, there may or may not be a module sheet.

- If you haven't written any code earlier (or run the recorder) you probably won't find a module sheet.
- Go to the PROJECT EXPLORER window of the VBA Editor, select your VBA Project (the name of the Excel workbook you are using), and then from top menu, INSERT a MODULE. One should appear, like a blank sheet of white paper, in the large window to the right.

Our first macro written from scratch



Expressions

- Definition: something that *returns a value* in VBA.
- For example,

```
3 + 2
zzz = 5
12
x = sqr(4)
x < 7
x = x + 3
```

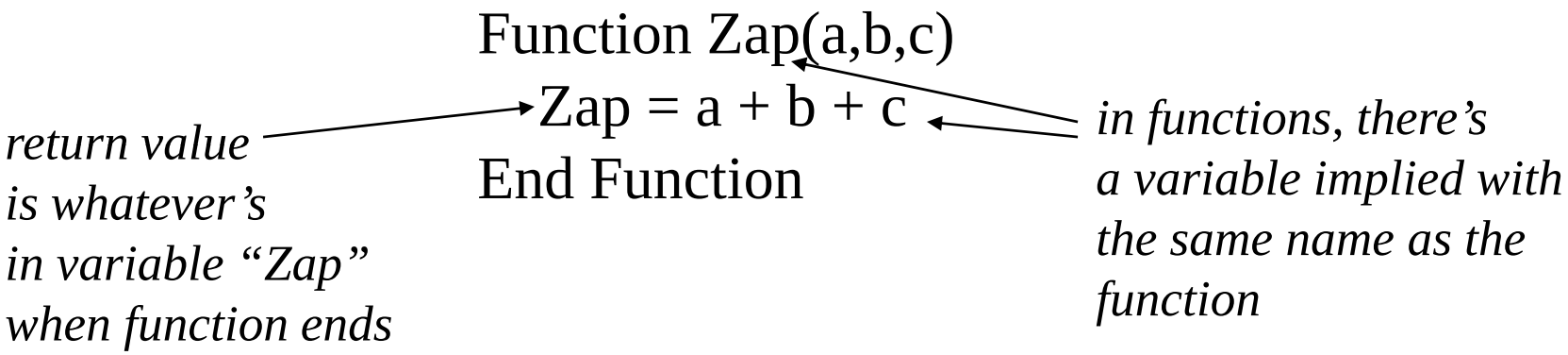
a VBA function, 'built in'

a logical expression (returns 'true' or 'false')

Types of VBA procedures

- **Macro**
  - a subroutine with an empty argument list
  - e.g. Sub Fred ( )
- **Subroutine** with nonempty argument list
  - can usually only be called by another subroutine, e.g. a macro (can't be run from the macro dialogue box, or the toolbar)
  - when Sub Fred calls Sub Ethel, Fred can pass arguments (values of something) to Ethel
- **Function**
  - a function is called from a spreadsheet cell, and after it runs there is a "return value" that is placed in the cell.
  - functions must have arguments passed to them.

A simple function

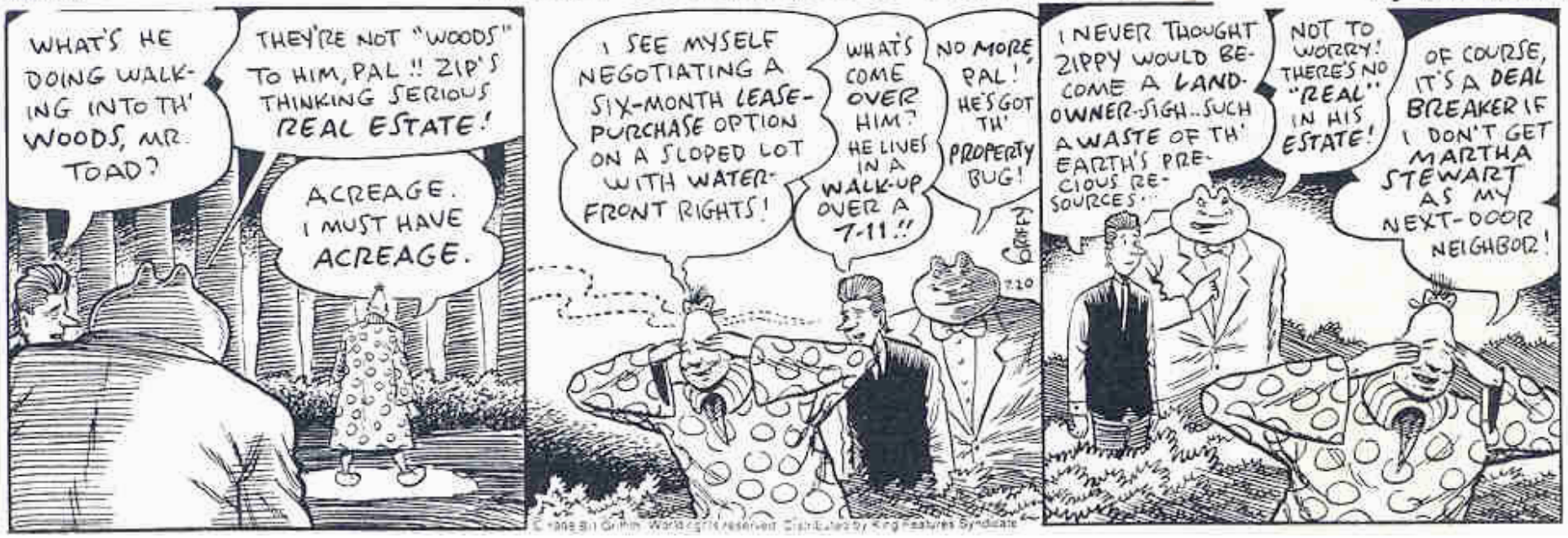


To invoke Zap, within a spreadsheet cell type the function, followed by the arguments, for example:

=ZAP(1,2,3)

Function ZAP will return the value 6.

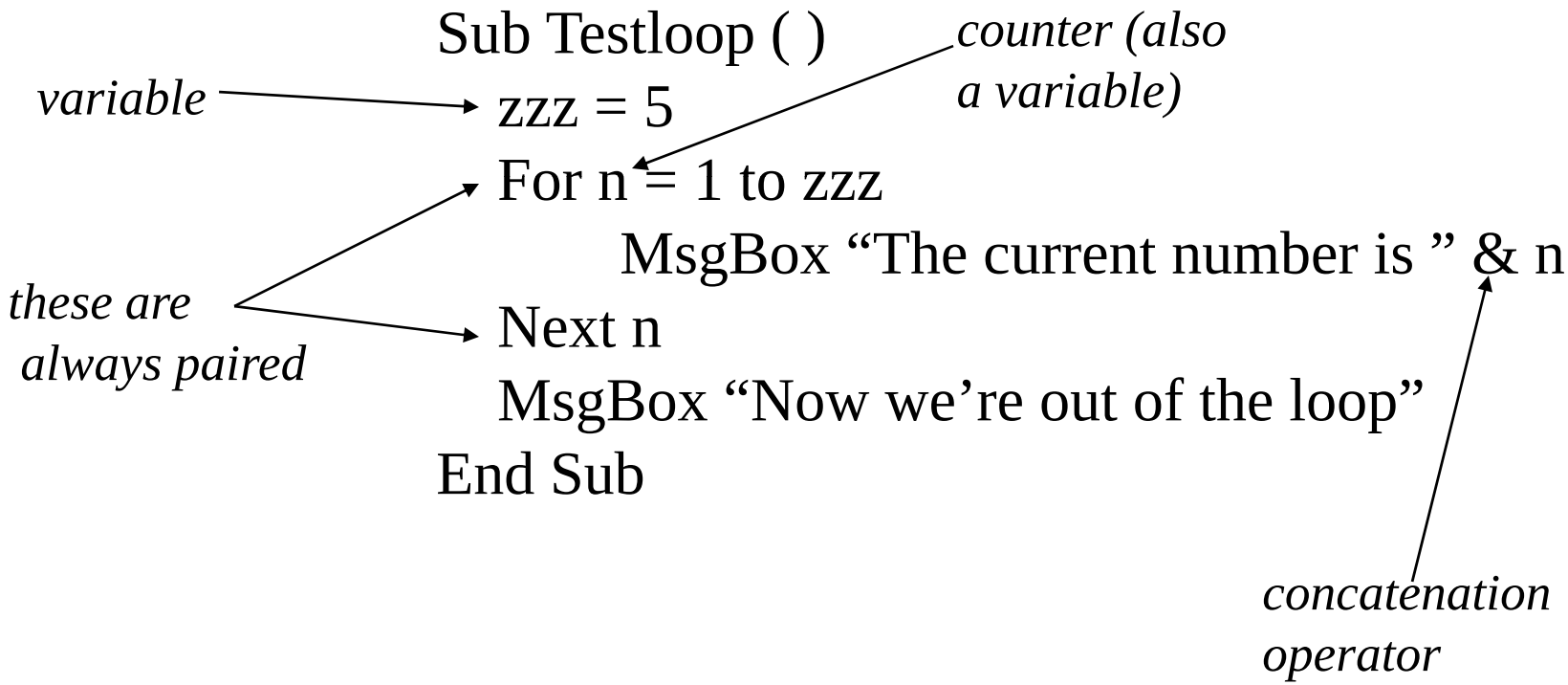
ZIPPY "PAINT EVERYTHING IN SEA FOAM" **By Bill Griffith**



Control Structure

- Without control structure, a macro runs from beginning to end.
- Control structure
  - Which statements execute? When/how often are they executed? Are they run at all?
- General elements of control structure
  - IF/THEN/ELSE
  - looping
    - FOR NEXT
    - DO WHILE
    - DO UNTIL
  - GO TO *(avoid)*

Example of control structure: "For-Next"



A little VBA style

- Use lots of comments.
  - Assume someone else will want to figure out what you've done. Make your program self-documenting.
- Indent your code to make it easier to follow.
- Don't be clever. Be clear.

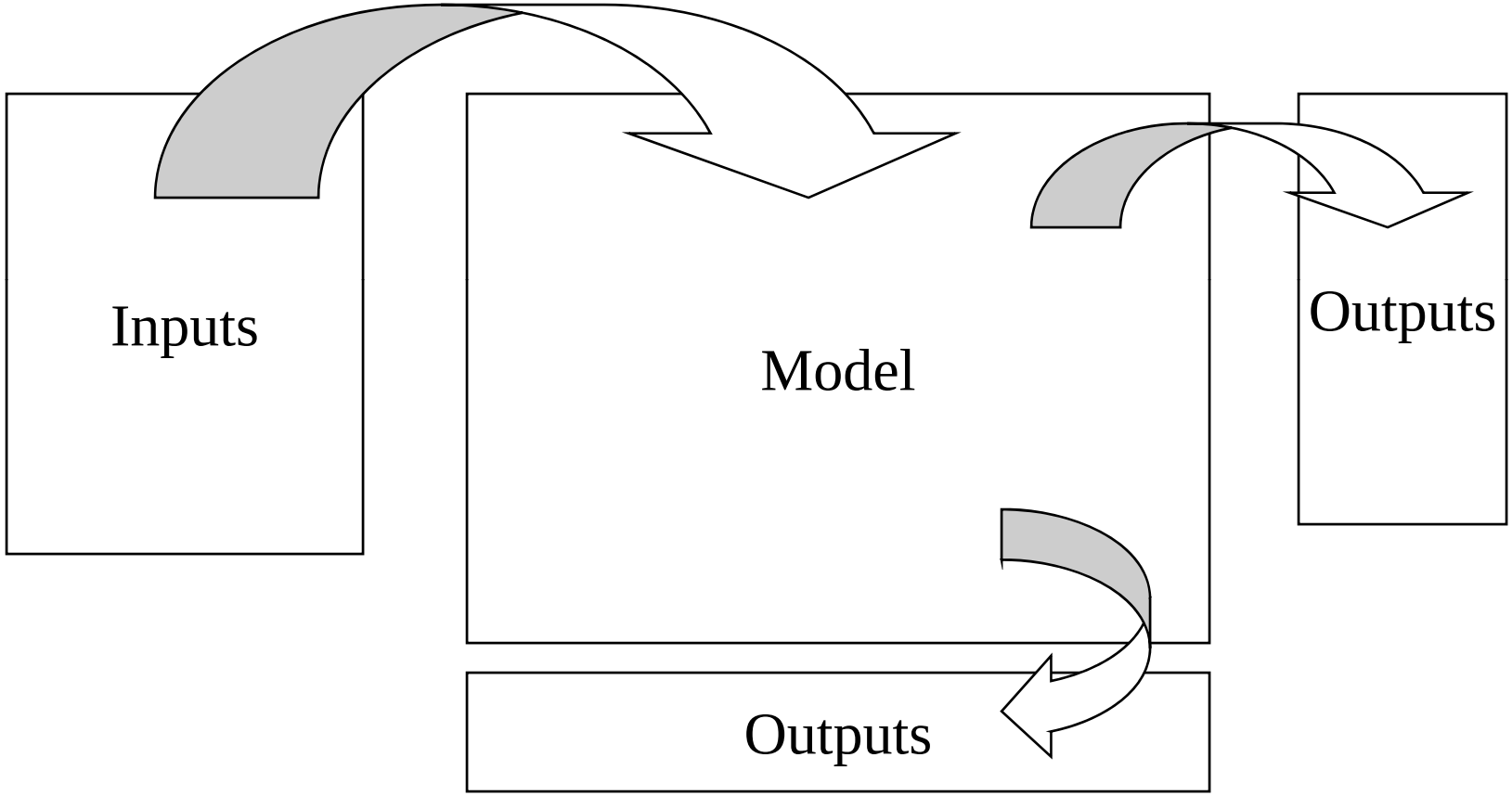
Our First Serious Exercise

- Modify our simple proforma model to handle many cases (investments)
- Key: instead of one column of inputs, let's have many rows of inputs. Each row is an investment.
- Run each row in turn through our (single) model.
- Let's have as many corresponding rows of output (present value, IRR, etc.)

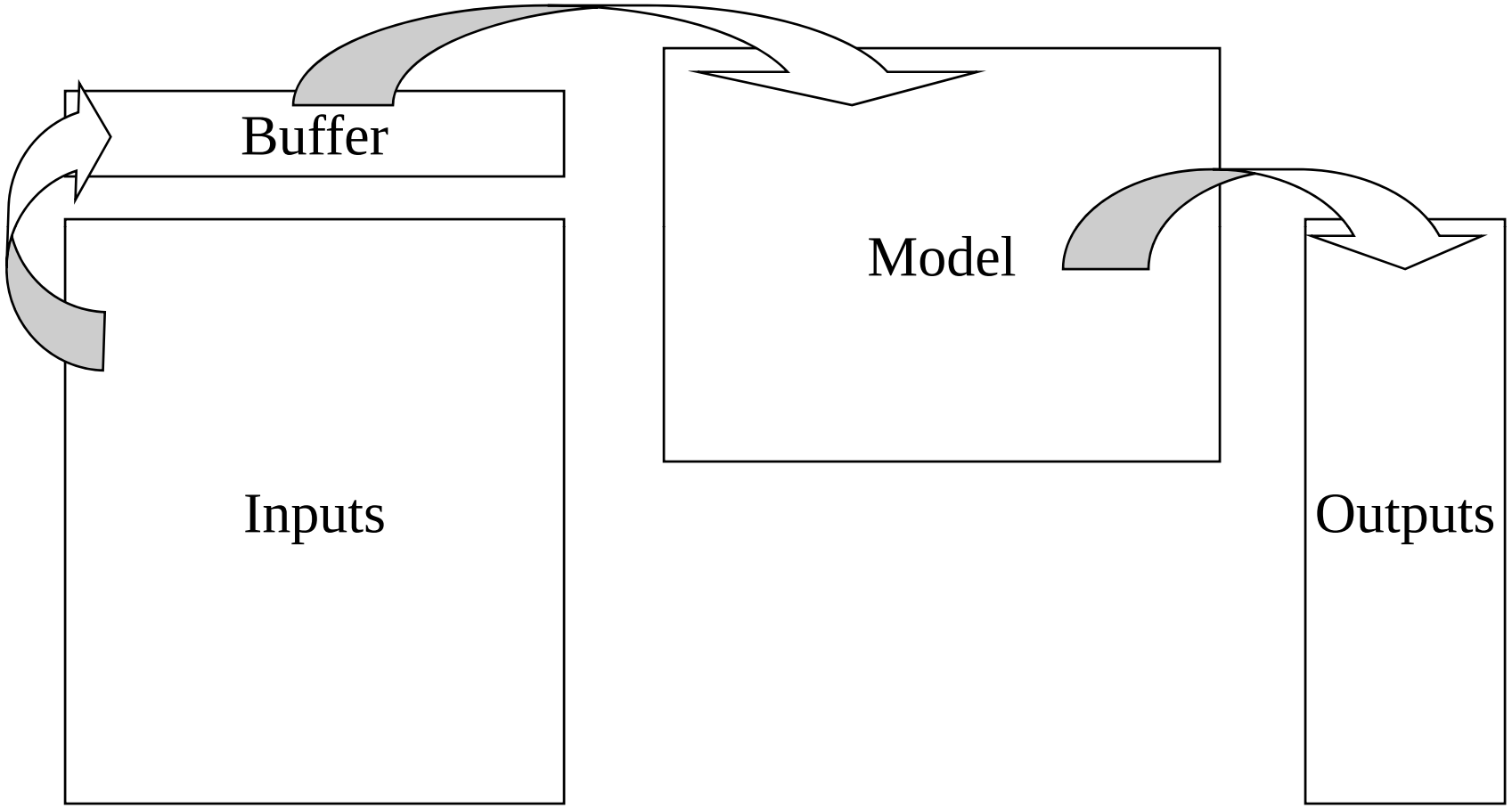
Three Stages of Spreadsheet Design (from above)

- **Think.** Map out our spreadsheet.
- **Implement.** Code carefully and clearly.
- **Debug.** Try to *break* the program. Every program ever written has errors.

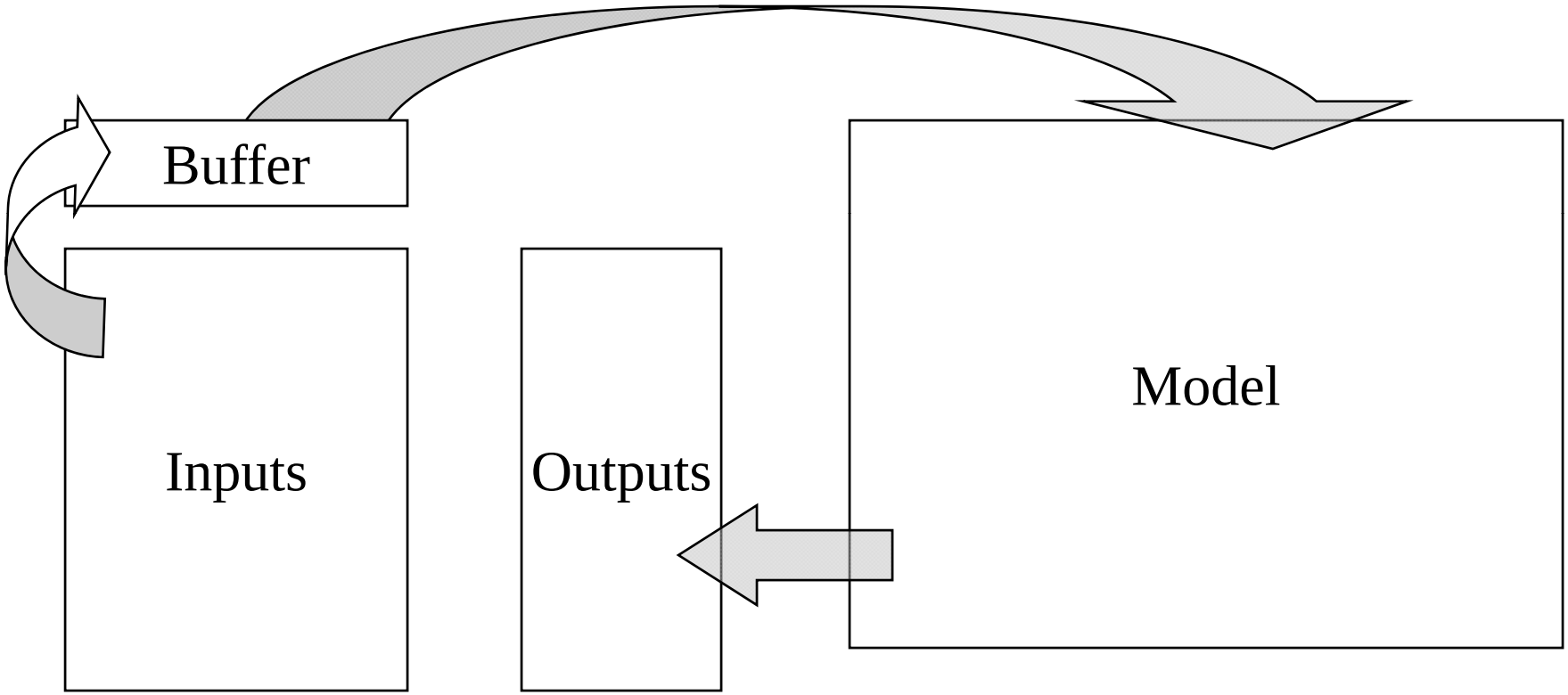
Earlier, we said: Good spreadsheet form is move left to right, and top to bottom.



Multiple cases: move each case (row), in turn, into the buffer; calculate the model; then move key outputs to corresponding output row



Rules were made to be broken.
(Why break them here?)



How do we figure this stuff out?

- Learn some VBA principles, and some code.
- Examine examples.
- Run the macro recorder.
- Use the online help.
- Browse books, and web resources.

***Principles*: Towards a Deeper Understanding of VBA**

The Grammar of VBA

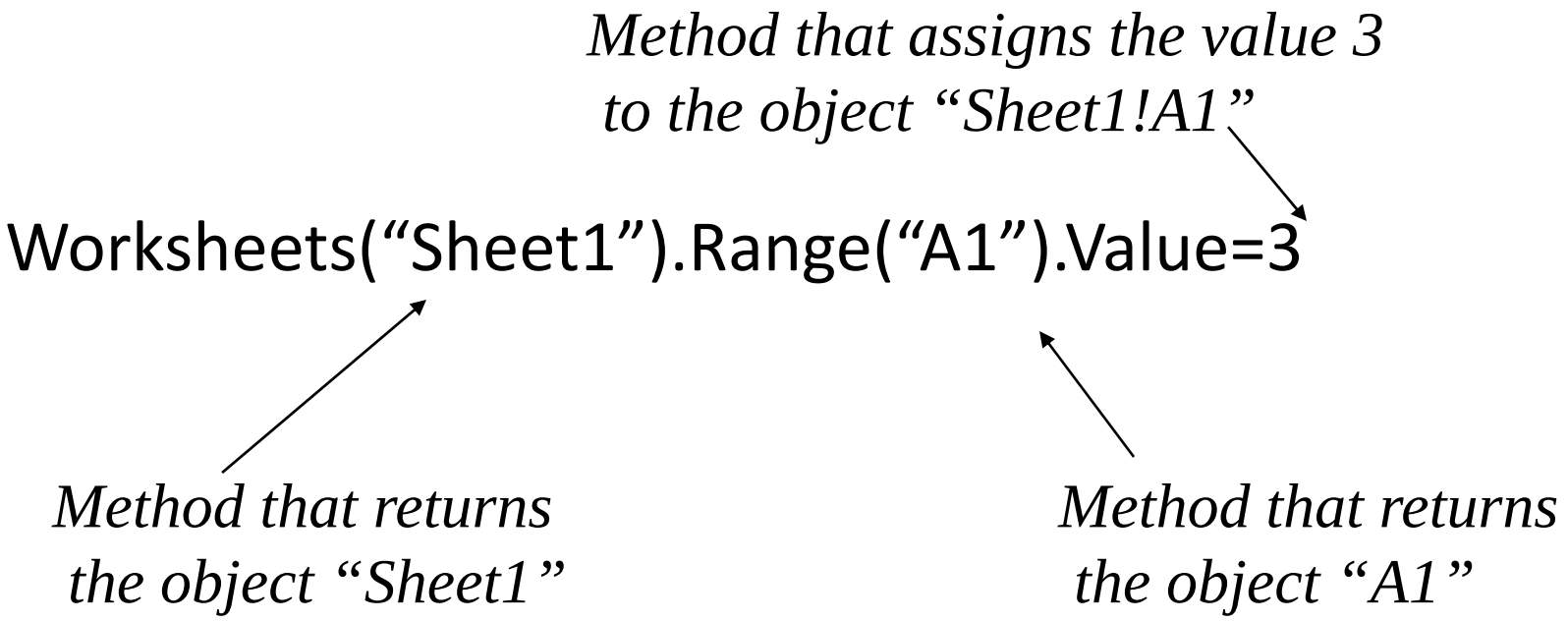
- **Objects** (nouns)
  - e.g. Workbook, Range, Font, Name
  - If you know an object's type or *class* you know what *properties* apply to it.
- **Properties** (adjectives)
  - e.g. Bold, Count, ColumnWidth, ActiveCell
- **Methods** (verbs)
  - e.g. PasteSpecial, Add, ClearContents, Copy, Edit, Sort

More VBA Grammar

- **Statements** (programming constructs)
  - EG For Next, Do While
- **Functions** (built in routines)
  - EG Abs, Rnd, Sgn, SolverLoad, Sqr
    - Note there are Excel functions and VBA functions; they are often similar but not identical.
- **Operators** (arithmetic, logical comparisons)
  - EG &, +, *, -, /, ^
  - EG And, Is, Not

What workbook? What sheet? What cell? What value?

- Example:



Notice that objects can be hierarchal.

Because we haven't ***specified*** the workbook, VBA will look in the workbook we are currently in.

Two ways to refer to (say) a particular range in VBA

```
Workbooks("Workbook 1.xls").Activate
Worksheets("Sheet 1").Activate
Range("Fred").Select
```

Three statements

Continuation " _"

```
Workbooks("Workbook1.xls"). _
Worksheets("Sheet 1").Range("Fred").Select
```

A single statement

Here's an interesting construction!

```
Worksheets("prob 3").Activate
    'copy a row of input into the buffer
    Range("XCASES").Rows(nnn).Select
    Selection.Copy
    ActiveSheet.Paste destination:=Worksheets("prob 3").Range("BUFFER")
```

Hmm. How did I figure out the "Rows" method of referring to a (*you guessed it*) row?

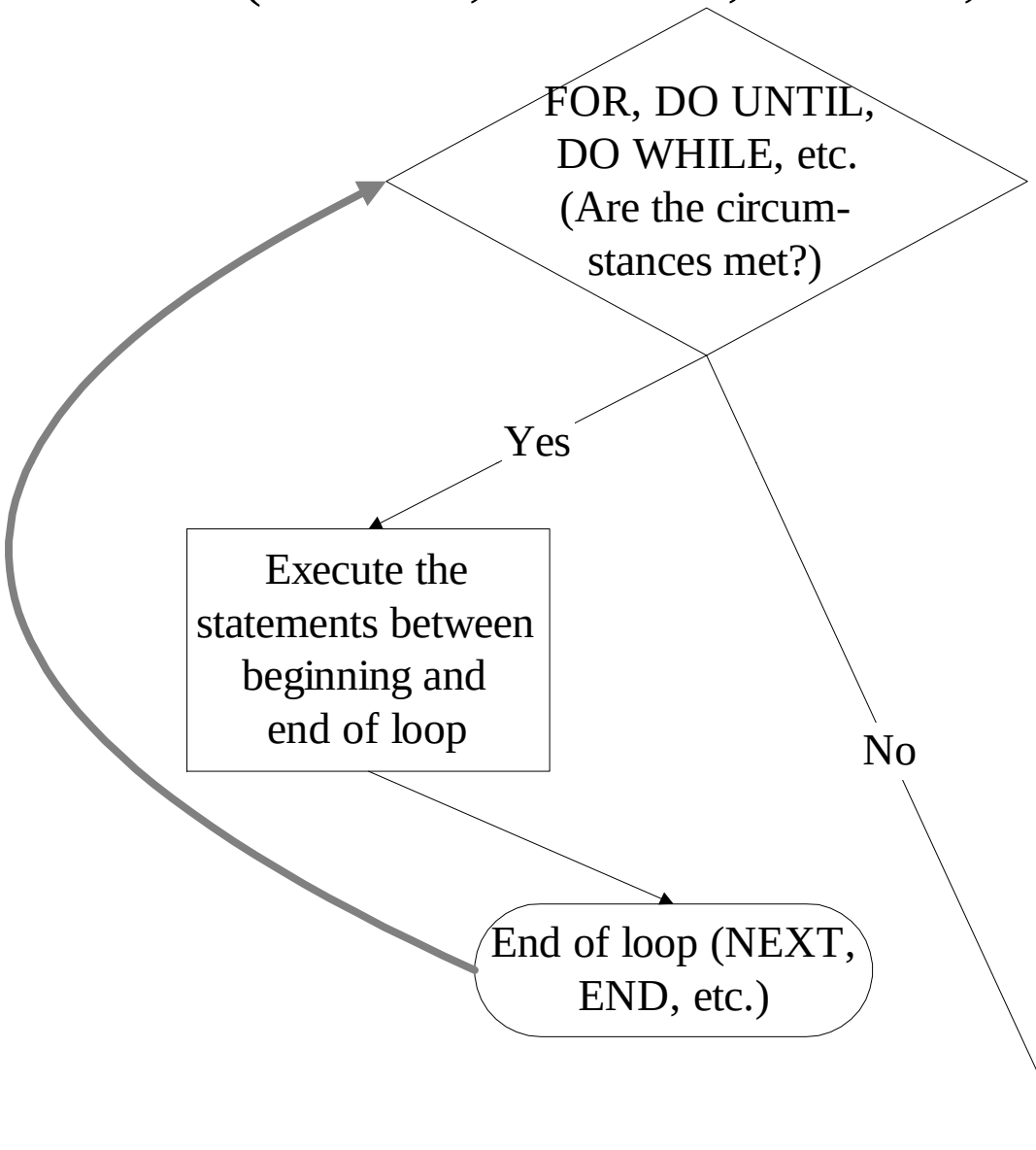
VBA Grammar

- Some words can be more than one part of speech, depending on context. EG range can be an object or a method.
- Like German: methods (verbs) come at the end.

Program Control

- If/Then/Else
- For Next
- Do Until
- Do While
- Select Case

The Logic of Looping Control Structures (For/Next, Do While, Do Until, etc.)



For Next

```
For x = 1 to 10
    Beep
Next x
```

- *Use For Next when you know exactly how many times to run the loop when you start.*
- *If you don't know this, use a Do Loop.*

Do While ... Loop

```
x = 0
Do While x < 10
    Beep
    x = x + 1
Loop
```

- *Use Do While ... Loop when you want to test a condition before you run the loop and then continue to run the loop while the condition is true.*

Do Until ... Loop

```
x = 0
Do Until x > 10
    Beep
    x = x + 1
Loop
```

- *Use Do Until ... Loop when you want to test a condition at the beginning of the loop and and then run the loop until the condition is true.*
- *NB: If the condition is true to begin with, the statements in the loop are not executed!*
- *Suppose we changed code above to "Do Until x < 10"*

If... Then... Else ...

- *Single-line syntax:*

```
If x < 3 Then y = 0
```

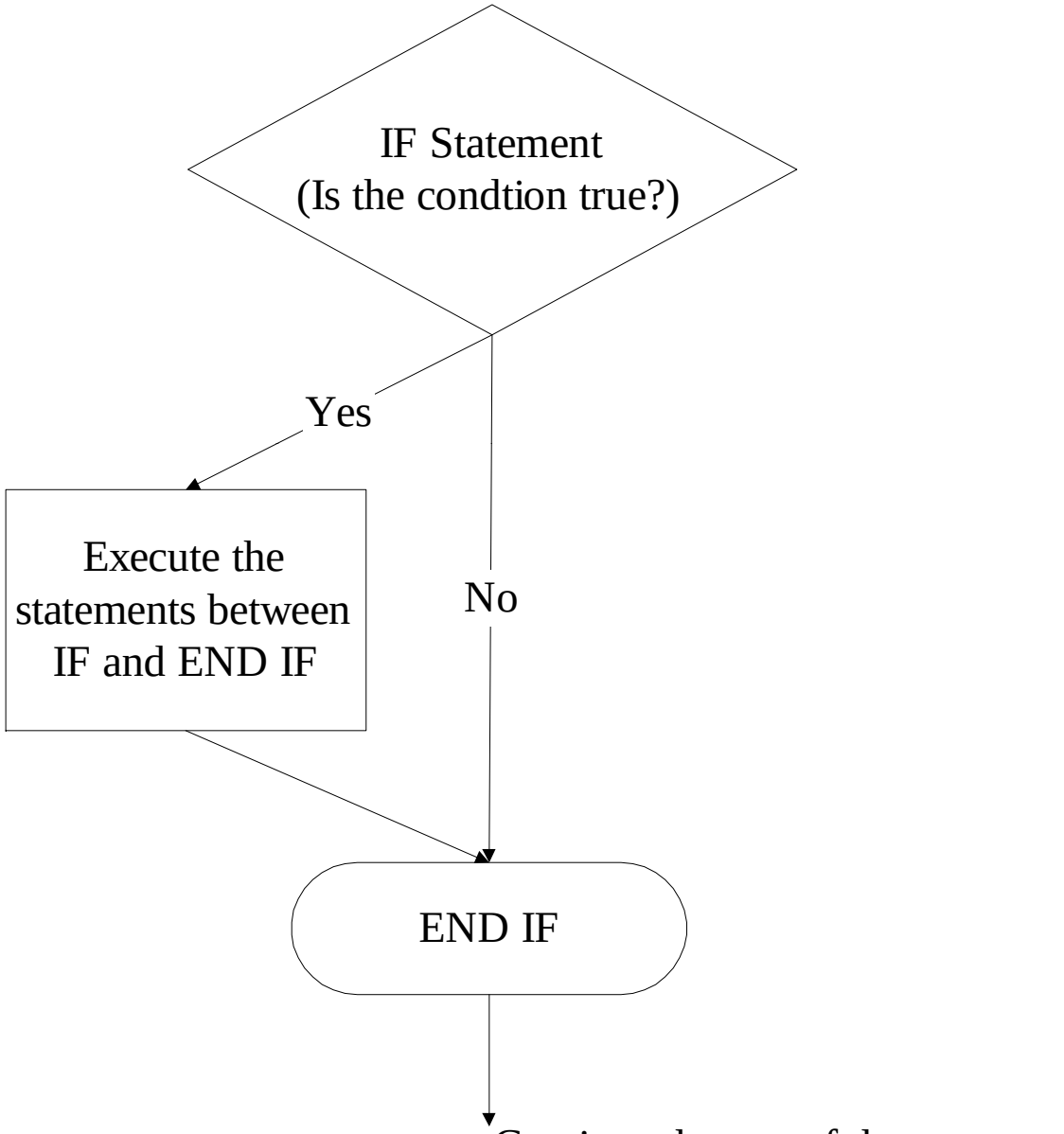
- *Block syntax:*

```
If x < 3 Then
    y = 0
    z = x + 1
End If
```

- *If, Then, Else:*

```
If x < 3 Then
    y = 0
Else
    y = 1
End If
```

The Logic of IF/THEN Control Structures



Elseif, and Select Case

- Suppose you have more than two alternatives to evaluate in an “If... Then...” situation
  - e.g. if you have several income classes, need to calculate taxes differently for each class
- You can “nest” If... Then... Else... statements (doable, but hard to follow.
- *You can use If... Then... Elseif*

```
If inc < 10000 then tax = inc * .2
Elseif inc < 20000 then tax = inc * .3
Elseif inc < 30000 then tax = inc * .4
Else tax = inc * .5
End If
```

Select Case

- *Select Case is similar to If... Then... Elseif*

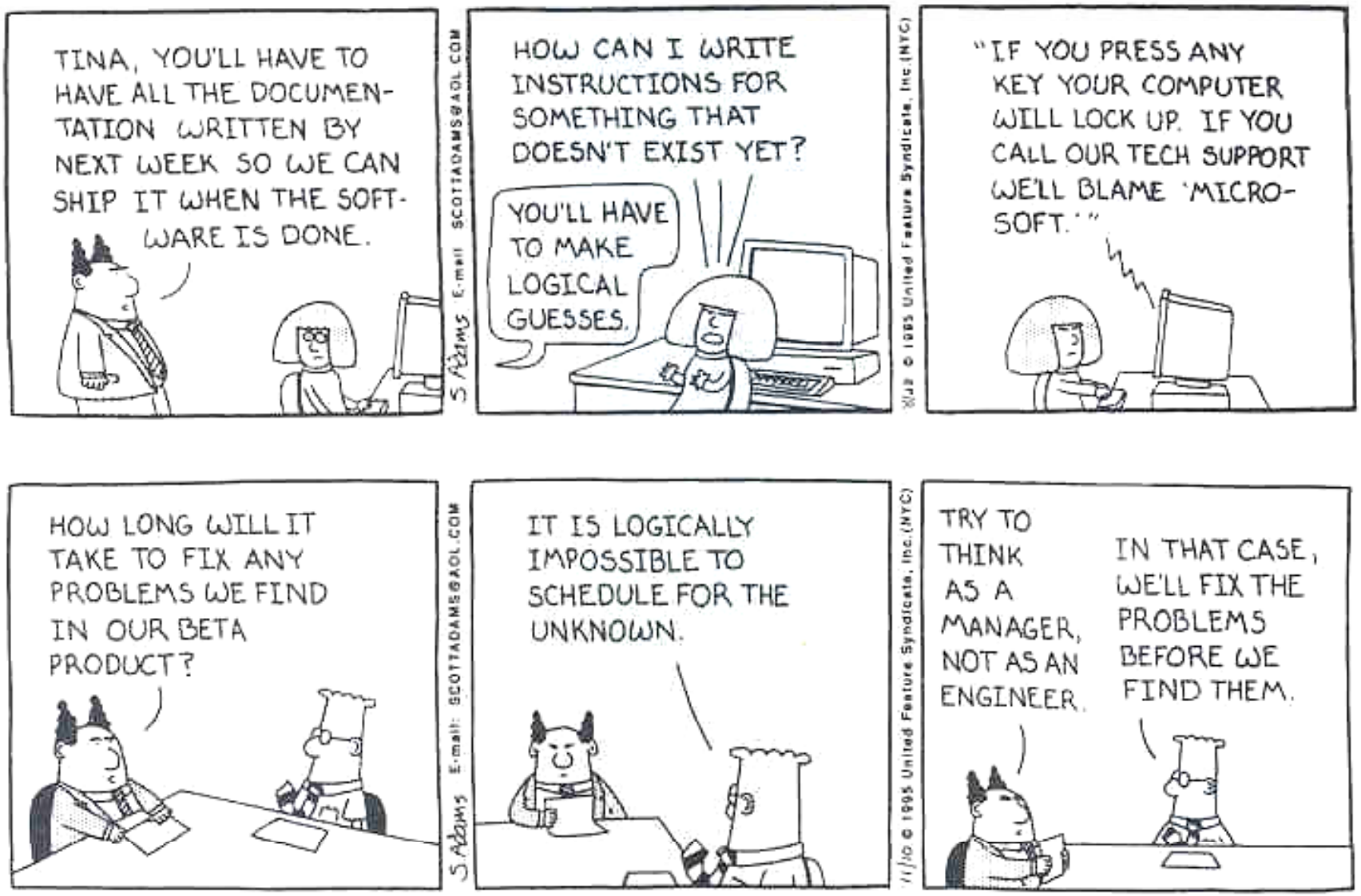
```
Select Case inc
    Case 0 to 10000
        tax = inc * .2
    Case 10001 to 20000
        tax = inc * .3
    Case 20001 to 30000
        tax = inc * .4
    Case Else
        tax = inc * .5
    End If
```

Functions

- Macros (subroutines) can be called in several ways, for example:
  - "Tools," "Macro," then select
  - Install buttons in worksheet or on toolbar
  - Macros can call other macros
- Macros don't pass arguments when called (except when a macro calls another macro)
- Functions, in contrast, are always called from within a cell; and one or more arguments are usually passed.
- Functions always return a value to the cell containing them.
  - Example: =SUM(1,2,3) returns the value 6 to the cell containing the formula.

Functions

- Microsoft provides a large set of pre-written functions (SUM, IRR, NPV, AVERAGE, and so on), but we can write our own.
- We'll write one today:
  - Compute internal rate of return for a series of cash flows
    - Unlike Excel's, our IRR function won't quit till it solves.
    - It isn't as efficient, and hasn't been thoroughly tested yet.


TINA, YOU'LL HAVE TO HAVE ALL THE DOCUMENTATION WRITTEN BY NEXT WEEK SO WE CAN SHIP IT WHEN THE SOFTWARE IS DONE.
S. Adams E-mail: SCOTTADAMS@AOL.COM
HOW CAN I WRITE INSTRUCTIONS FOR SOMETHING THAT DOESN'T EXIST YET?
YOU'LL HAVE TO MAKE LOGICAL GUESSES.
"IF YOU PRESS ANY KEY YOUR COMPUTER WILL LOCK UP. IF YOU CALL OUR TECH SUPPORT WE'LL BLAME 'MICROSOFT.'"
HOW LONG WILL IT TAKE TO FIX ANY PROBLEMS WE FIND IN OUR BETA PRODUCT?
S. Adams E-mail: SCOTTADAMS@AOL.COM
IT IS LOGICALLY IMPOSSIBLE TO SCHEDULE FOR THE UNKNOWN.
TRY TO THINK AS A MANAGER, NOT AS AN ENGINEER.
IN THAT CASE, WE'LL FIX THE PROBLEMS BEFORE WE FIND THEM.

How to turn off the monitor in VBA

- When we run a macro in VBA, the screen jumps as if we're carrying out the procedure using keyboard and mouse, except very fast.
- This kills efficiency, because changing the screen so much takes lots of computing power.
- To shut off, put the following instruction at the beginning of your macro:

```
Application.ScreenUpdating = False
```

Spaces sometimes matter in VBA

- Make sure there's a space before and after the concatenation operator, & . Otherwise VBA may interpret it as a type-declaration character.
- The VBA line continuation sequence is actually two characters: a space, followed by the underscore _

How to run a macro using a button in a cell

- Get into the worksheet you want to contain the button.
- Right-click any toolbar, and select *Forms* from the menu.
- Click once on the Button tool on the Forms toolbar. Then move the cursor to the cell you want to contain the button.
  - Excel immediately presents you with a list of macros, so you can choose which one to assign to the new button.
  - After you've created it, you can right click on the button and change the text, font, etc.
- Once it's finished, click on the button to run the macro.

How to run a macro using a button on the toolbar

- *Tools*
- *Customize*
- *Commands*
- *Macros*
- drag *Custom Menu Item* to toolbar
- right-click item to assign macro, give it a title, format the menu item.
- Once you're done, click on the toolbar button to run the macro.

Defining Variable Types with DIM Statements

- Variables in VBA (and other environments, includng Excel) can be many types:
  - Integers
  - Floating-point numbers (for decimals and fractions)
  - Characters
- They can also be many sizes
  - A single number, or scalar
  - An array, or matrix, of a given size (e.g. a 10x1 vector, or a 5000x10 array)
- VBA is usually pretty good about figuring out variable types, and sizes, on the fly.

Defining Variable Types with DIM Statements

- VBA is usually pretty good about figuring out variable types, and sizes, on the fly.
- But it can be more efficient, and also good programming practice, to predefine variables before they are used. Examples:
  - DIM I as Integer
  - Dim X(3, 4) As Integer
  - Dim Y(1 To 5, 4 To 9, 3 To 5) As Double
  - Dim BirthDay(1 To 10) As Date

Going forward

- Lifetime learning: Set aside 20-30 minutes every week or two to browse an Excel manual or the online help, or to study someone else's models.
- Start with today's material.
- Take RE 631, Computer Applications. (If not, collect materials from the website).
- Examine other people's models. Learn from them. Drop the junk.
- Online help, web sources, browsing your favorite bookstore.