

# ROI of Automation

Justifying the purchase of test tools...  
...and related random thoughts

Ron McClintic  
McClintic Services, Inc.

# Today's Content

- Review definition of ROI
- How ROI is supposed to be used
- How does this relate to software testing?
- Do we really do ROI or compare I's?
- Soft versus hard dollars
- Calculating ROI
- Demo of an ROI calculator...or 2
- Conclusions

# Disclaimer

- I am NOT anti-automation
- I am FOR accurate analysis
- I am FOR analysis done for the right reason
- I have:
  - Sold tools
  - Bought tools
  - Consulted on selection
  - Used tools
  - Managed the bottom line

# ROI Defined

- Return on Investment
- Return = Increased value or sum of values
- Investment = Cost or sum of costs
- Capital versus expense
- $ROI = V / (C_c + \sum C_v)$
- Typically annualized

# ROI Formula - Incremental

$$\text{ROI} = \frac{\text{NPV}(V_f - V_c)}{C_c + \text{NPV}(\sum C_v)}$$

Discounted difference between ending and current values, divided by Capital cost plus the discounted stream of ongoing costs.

# Internal Rate of Return - IRR

The return on investment before outside costs (e.g. taxes, interest)

- Used for Go/No go decisions
- Investment in test tools will not only compete against manual testing but also against every other investment choice – like a new product

# Opportunity Cost

The cost related to the second best choice.

- The opportunity cost of automation is the benefit you would have earned manually testing instead of creating scripts
- Testing delayed while creating automated scripts

# How high is the bar?

- Incremental ROI must be significant
- ROI of tools must be higher than other QA choices
- IRR must compare favorably with ALL other spending choices

# Relating to Software Testing

- Who would ever be this exact?
- Can we even quantify R and/or I?
  - Initial cost? **YES**
  - Ongoing expense? Maintenance **YES**, initial training **YES**, what about other ongoing costs?
  - Return? Who can calculate today the return on investment of their current test methods?
  - IRR target can be found in annual report (or ask the CFO)

# Calculating Investment

- Acquisition cost
- Hardware
- Maintenance contracts (rule of 70)
- Training
- Others?
  - Cost to maintain scripts?
  - Cost of creating scripts?
  - Training new people? Supplemental training?

# Calculating Return

- There is no revenue. So what do we measure?
- Incremental return above and beyond the return from manual testing.
- What is return on manual testing?
- Hours spend testing?
- Defects found?
- Speed to market?
- Meet regulatory needs?

# Problems Calculating Return

- Value of faster runs?
- Value of more execution cycles?
- Is an automated test really more valuable than a manual test?
- More expensive to create?
- Tools expensive so return must be higher?
- Other issues?

# Ideal Return

- You have to reduce it to dollars.
- Either the dollar value of a defect free release
- And/or the dollar value of faster to market
- And/or peace of mind of 100% regression
- Theoretically this is in the Charter doc 😊
- But these are very hard to calculate.

# Calculator Demo

- <http://www.automatictestingframeworks.com/>
- The best ROI calculator I have seen
- Dorothy “Dot” Graham, co-author of “Software Test Automation: Effective use of test execution tools”
- Rational’s Calculator

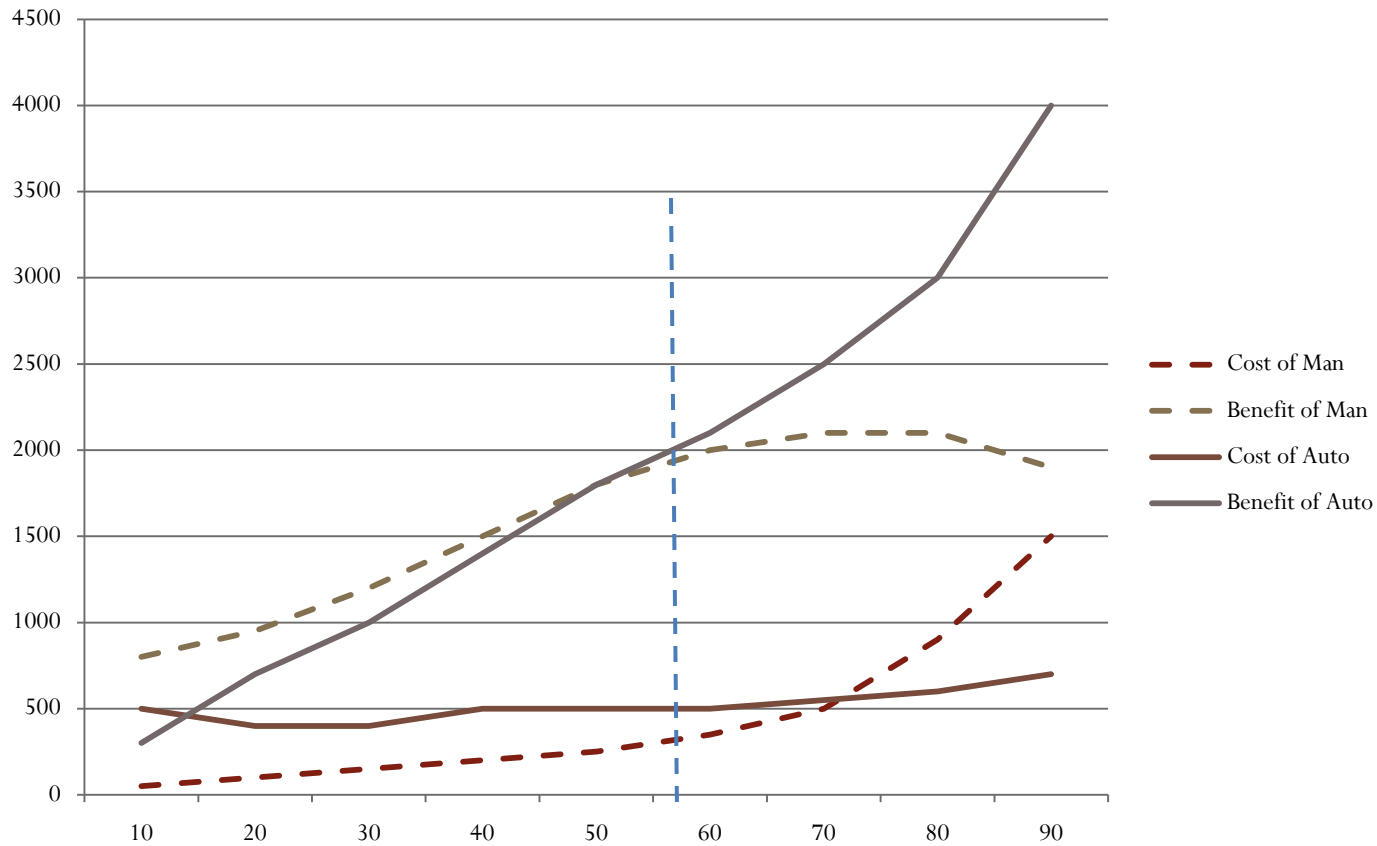
# Build your own (1)

- $E_m$  = Time it takes to execute the all cycle manually  
 $M_m$  = the sum time of updating the manual testing documentation after each run, per cycle  
 $N$  = number of cycles that need to be executed  
 $A_u$  = Time it takes to automate the cycle  
 $E_a$  = time it takes to execute full cycle automatically  
 $M_a$  = time it takes to modified the automatic framework on changes between cycles.
- So, according to time factor, automation is suitable if:  
$$N * (E_m + M_m) > A_u + N * (E_a + M_a)$$

## Build your own (2)

- $P_m$  = regression cost on creating and executing manual testing (include number of workers, working hours, working place and terms)  
 $P_{mu}$  = update regression cost of manual testing
- $P_a$  = regression cost of automating the cycle (tools price, developer working hours and terms, automation station cost)  
 $P_{aex}$  = regression cost of executing the cycle  
 $P_{au}$  = update regression system between cycles (developer hours, automation station cost )  
So according to cost criteria, automation is worth if
- $N * (P_m + P_{mu}) > P_a + N * (P_{aex} + P_{au})$

# Sweat Spot



# Survey Question

- Which would you choose.....
  - \$600k tool investment, \$50k training
  - \$150k training, use collection of shareware tools

# Conclusions

- Given the cost to create automation scripts, we need to know the real tests to run.
- Using shareware makes the decision easier.
- You can use the calculator to justify a purchase.
- I am not confident there is ever a greater ROI in automation than in manual testing – with a couple exceptions.
- Test tools purchases are competing against the entire company's possible investment choices.

# Automation Framework

For higher management, its a tree that grows out of nowhere with zero investment in resource and time - and solves all testing related issues overnight magically.

- For fresh automation developer, its an argument for buying little more time in an obviously impossible schedule.
- For moderately experienced automation developer, its a set of useless guidelines leads wants to believe are being followed and some reusable class/function/tool which are actually useful.
- For highly experienced automation developer, its a grand tool s/he created but very few people ever really used.
- For client, its another trick of vendors to steal more money.
- For other developers - "huh ?? what the hell is that ?"