Easter 2010

DEVELOPING COMMERCIAL SOFTWARE

Product vs. Technology

- Technology is written for the sake of the technology
  - Research
  - Prototyping
  - Platform Development
- Product is written to be sold
  - Has an identified market
  - Output is something which can be sold

The Commercial Software Team

The SQL Response 2.0 Project Team

#	Role	Responsibility
1	Scrum Master/Project Manager	Scrum master, coordinates backlog.
1	Product Manager/Owner	Commercial input. Represents users.
1	Usability Engineer	Designs visual aspects of product.
1	Technical Author	Responsible for all text in the product.
5	Software Engineers	Architect and develop the software
4	Software Testers	Ensure the software complies with user stories

This is just one team within Red Gate but the ratios are fairly standard.
As product and project needs change we will vary the number of people and ratios.

Stages of a Project

Stage	Length	Output	People Involved
Research Phase	1 month – 5+ years	Business case approval	**PM**, UX
Pre-greenlight	2 weeks – 2 months	Project Approval	**PM**, SM, UX, Dev
Greenlight	1-4 weeks	Backlog	**SM**, PM, UX, Dev, Test
Pre-EA	n Sprints	1st EA Build	**SM,** Dev, UX, Test, PM
EA Program	m Sprints	m EA Builds Beta 1	**SM,** Dev, UX, Test, PM
Beta	4-8 weeks	Release Candidate 1	**SM,** Test, Dev, UX
Release Candidate	2-4 weeks	Release Build	SM, **Test,** Dev
General Availability	-	-	-
Research Phase	1 month – 5+ years	Business case approval	

Product Management

- NOT the source of all ideas
- Responsible for pulling together desperate sources of information & collating into a roadmap
  - Often has P&L Responsibility for a product
- Differs from Marketing
  - Product Marketing Manager TALKS
  - Product Manager LISTENS

Research Methods

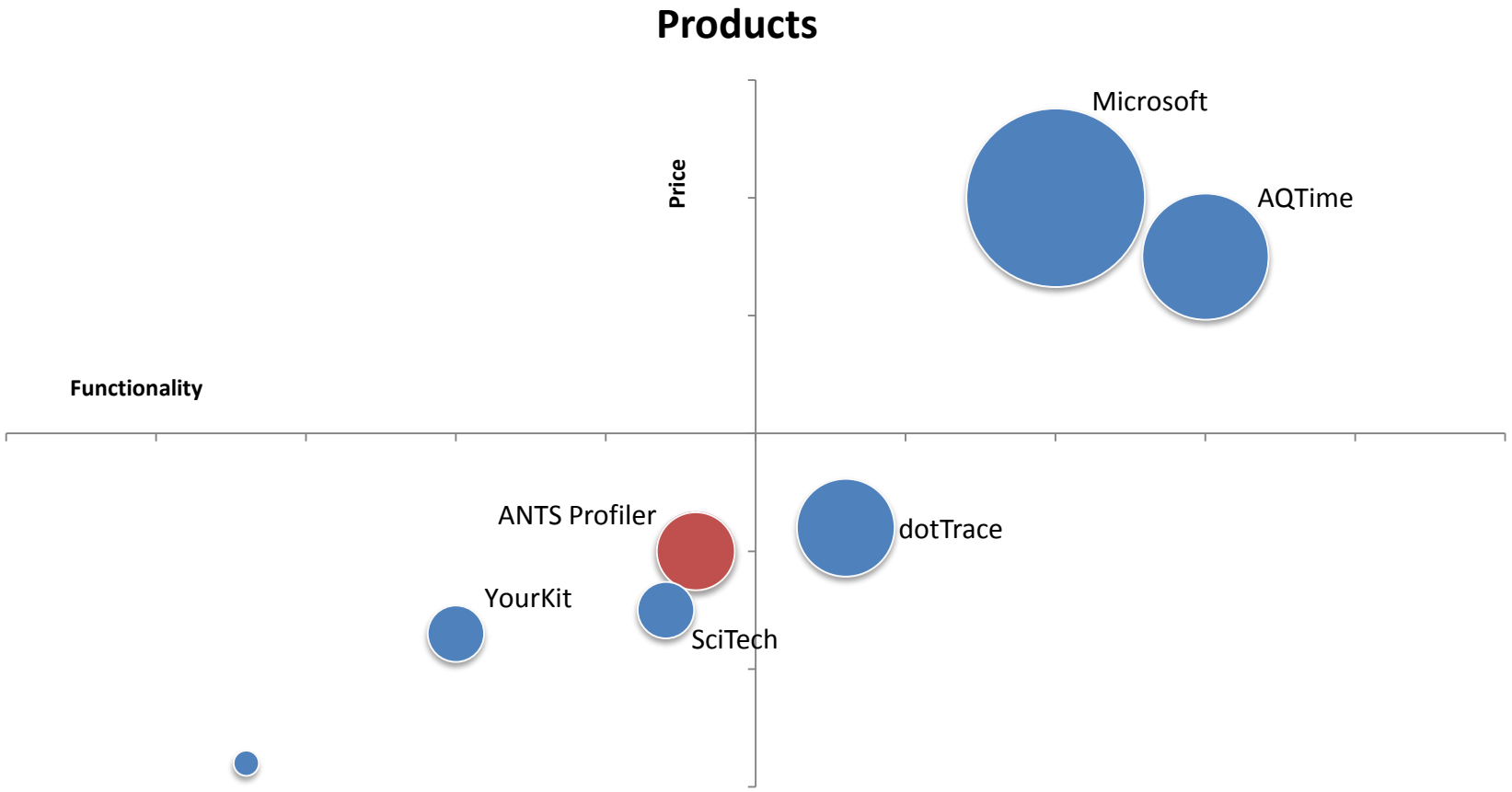
- Customer Visits
- Surveys
- Competitive Analysis
- Customer Feedback
- Support Requests
- Analyst Reports
- Partner Customer Research Reports
- Corporate Annual Statements
- Win/Loss Analysis
- Any other way of getting data on the market!

ANTS PROFILER V4

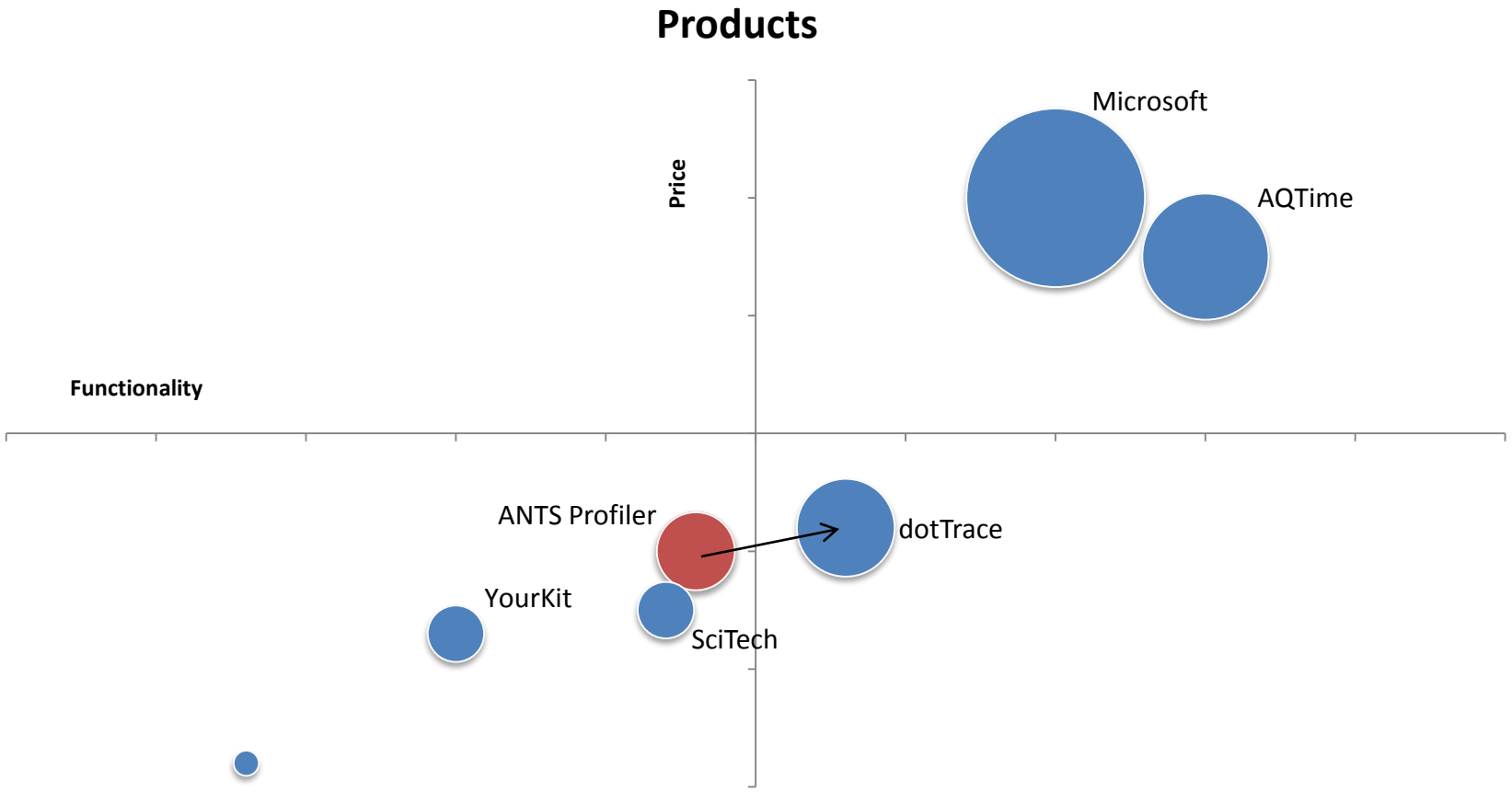
Win/Loss Analysis

Win	**Loss (Bought competitor)**	**Loss (No Purchase)**
Ease of use	Speed	Trial solved my problem
Price	Ease of use	No native code support
Good support	Price	Too expensive
Used it at a previous job	Used it at a previous job	Looking to purchase soon
	Supported platform X	
	Could not get it to work	
	Has Attach to Process	
n=20	*n=15*	*n=45*

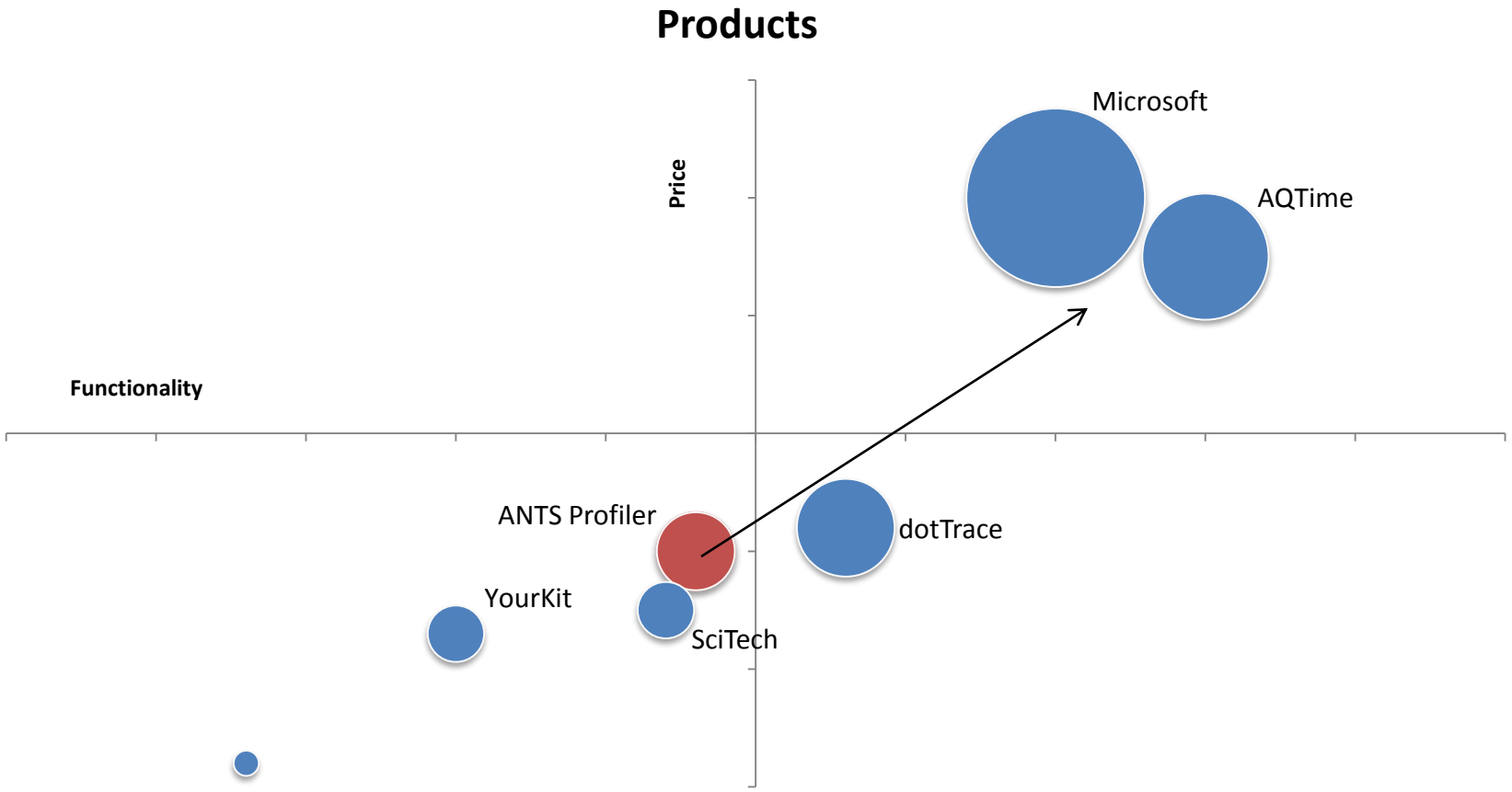
Competitive Positioning



Competitive Positioning



Competitive Positioning



Surveys

- Use to check your Hypothesis
- Cheap
- Hard to be statistically significant
  - We normally try to get 300-500 responses

ANTS Profiler v4: Key Findings

- Speed of Performance Profiler
- Attach to existing process so the user can look at part of a run
- Memory profiler could not deal with large amounts of data
- Memory Profliler overhead was too large
- Some customers preferred the competitors GUI

The Business Case

- Clear idea of what we wanted to achieve
- Is it worth doing it?
  - Need Revenue Projections
  - Need Cost Projections

Modelling the Business

- Projecting Revenue directly is tricky and error prone
- Build a simple model of the business
  - Keep it simple
  - Use numbers you can have some control over

Modelling the Business

- Downloads: Number of people trying out the tool
- Conversion Ratio: The percentage of these people who purchase the tool
- Average Transaction Value: The average spend of a customer

Revenue = Conversion Ratio x Downloads x Average Transaction Value

Average Transaction Value $\propto$ *Price*

Modelling the Business

- *Use simple cost models unless something more sophisticated is needed*

$$Monthly\ Project\ Cost = Number\ of\ People \times 23 \times \$1{,}000$$

$$Cost\ of\ Project = Monthly\ Project\ Cost \times Number\ of\ Project\ Months$$

Return on Investment

- Given identical risk profiles and the following projection of cash flow should we invest Project A, Project B or neither of them?

Year	Project A	Project B
1	(3,000,000)	(3,000,000)
2	0	1,000,000
3	500,000	1,000,000
4	1,000,000	1,000,000
5	1,500,000	1,000,000
6	2,000,000	1,000,000

Future Value

- If you have £100 today how much is that worth in 2 years time?

$$Future\ Value = Present\ Value\ \times (1 + i)^t$$

i = Interest rate

t = Number of time periods (years)

$$Future\ Value = 100 \times (1 + 0.05)^2 = 110.25$$

Discount Cash Flow/Net Present Value

- I will offer you a contract whereby I will pay you £100 in two years time. How much is that contract worth today?

$$Discounted\ Present\ Value = Future\ Value \times (1-d)^t$$

where $d = i/(1+i)$

$$Discounted\ Present\ Value = 100 \times \left(1 - \left(\frac{0.05}{1+0.05}\right)\right)^2 = 90.70$$

Discount Cash Flow/Net Present Value

- For a cash flow:

$$Net\ Present\ Value = \sum_{t=1}^{N} \frac{C_t}{(1+i)^t}$$

i	Project A	Project B
5%	1,065,198	1,266,168
10%	391,728	718,897)

Internal Rate of Return

- For a cash flow:

$$Net\ Present\ Value = \sum_{t=1}^{N} \frac{C_t}{(1+i)^t}$$

- If we have costs and expected return then set NPV = 0 and solve for *i*

$$IRR(i) = -3{,}000{,}000 + \frac{1{,}000{,}000}{(1+i)^1} + \frac{1{,}000{,}000}{(1+i)^2} + \frac{1{,}000{,}000}{(1+i)^3} + \frac{1{,}000{,}000}{(1+i)^4} + \frac{1{,}000{,}000}{(1+i)^5} = 0$$

Internal Rate of Return

- Secants Method:

$$r_{n+1} = r_n - \frac{r_n - r_{n-1}}{IRR(r_n) - IRR(r_{n-1})} IRR(r_n)$$

- Approximate r_0 and r_1 then repeat until solution converges

Iteration	r0	r1	r2	r3	r4	r5	r6
Project B	25.00%	35.00%	18.38%	20.28%	19.87%	19.86%	19.86%

Pitfalls with NPV, DCF and IRR

- Negative NPV projects might still be worth doing
- NPV calculations compound discount rates
  - Do not adjust for risk
- DO NOT discount known future costs
- NPV is an absolute number
  - Represents accretive value to shareholders
- IRR is expected return on capital
  - Does not include the cost of capital itself

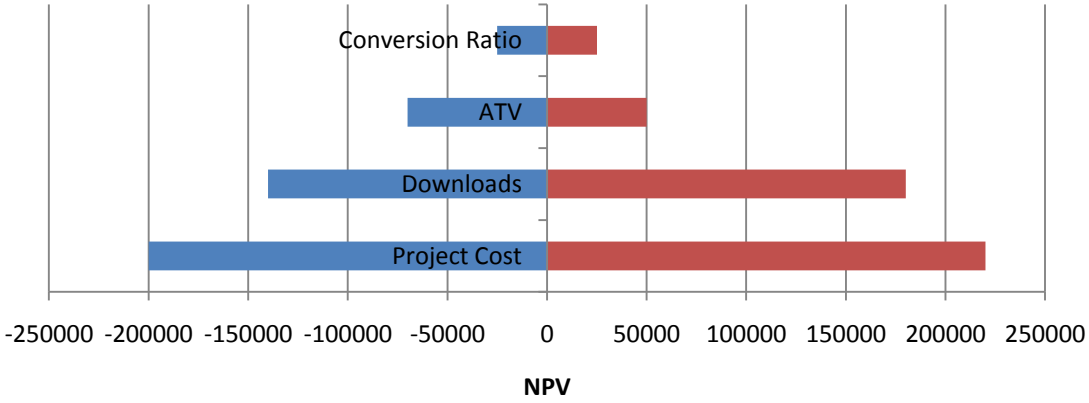
Dealing with Uncertainty

- Lots of uncertainty in model
- Use Monte-Carlo Analysis
  - Replace single valued inputs with PDFs
  - Run the model thousands of times collecting output values

- For example:
  - Project Length: U(9,18)
  - Accretive Conversion Ratio: N(0.05, 0.03)
  - Additional Number of Leads: U(1000,5000)

Sensitivity Analysis

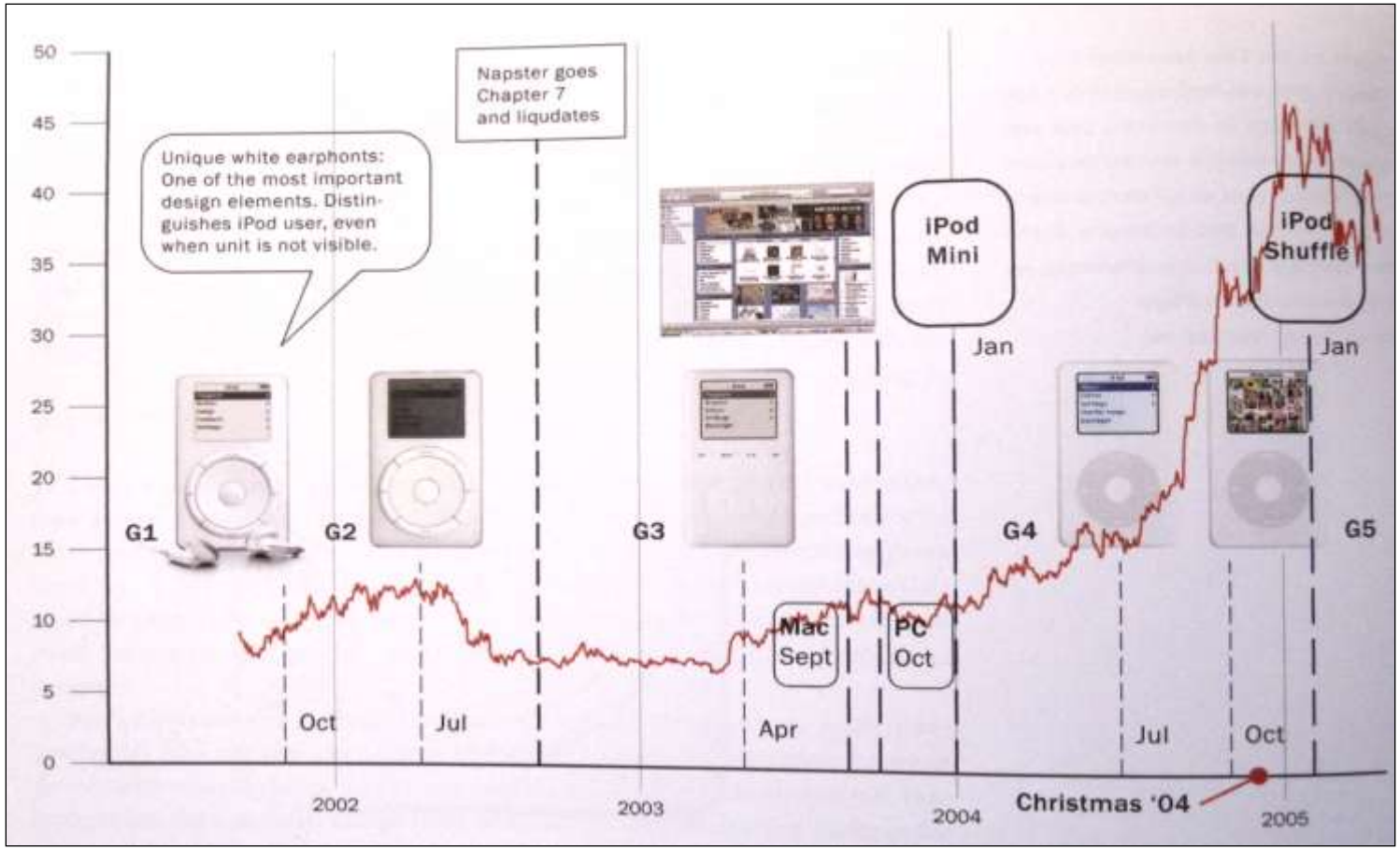
- What is the effect of each assumption?
- Use Monte-Carlo Analysis
  - Replace single valued inputs with PDFs
  - Run the model thousands of times, but only vary a single pdf, collecting output values
- Tornado chart:



USABILITY

????

The iPod



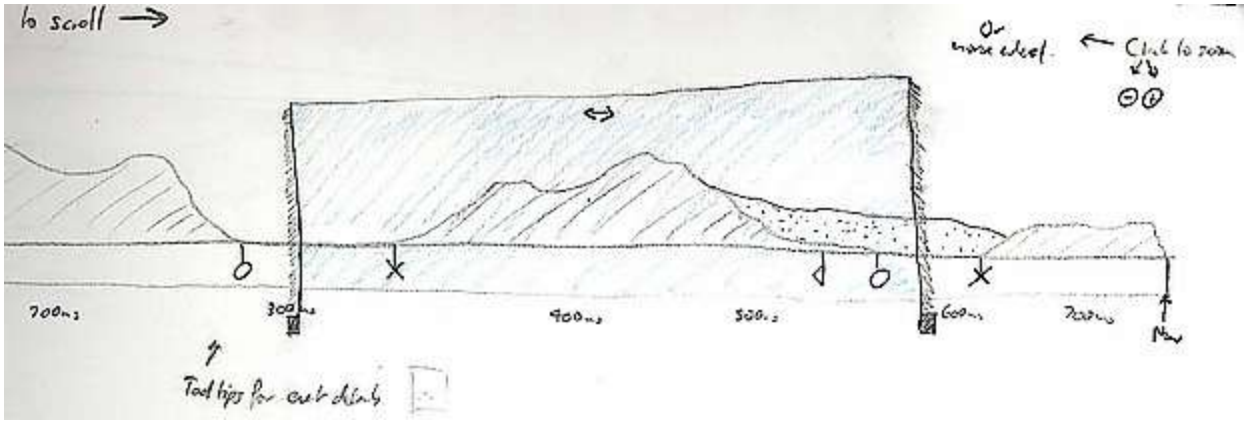
Bill Buxton, Sketching User Experiences, p48, 2007

Improving Ease of Use

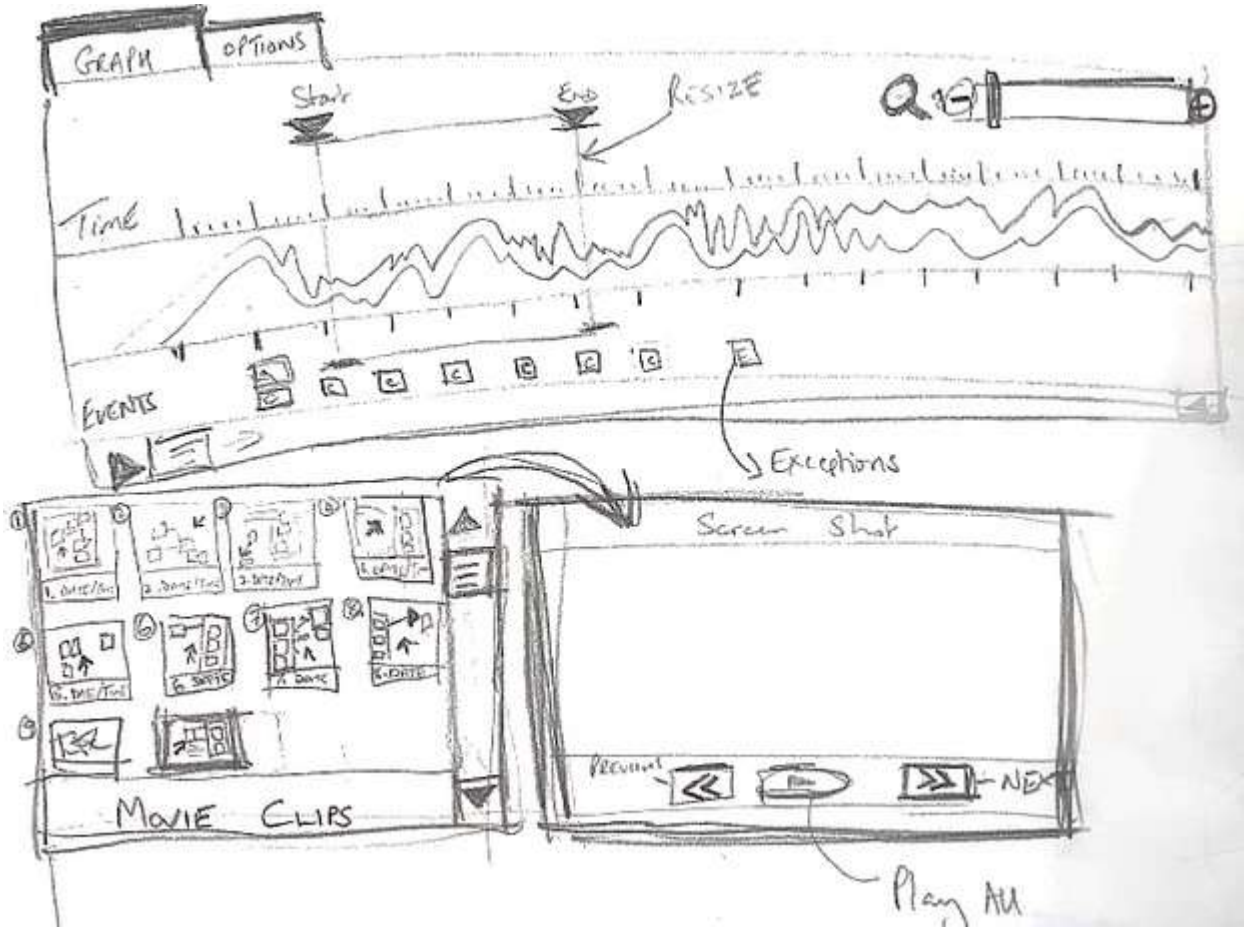
- Had idea of how to solve speed issue
- Wanted to make sure we kept all of our options open

Improving Ease of Use

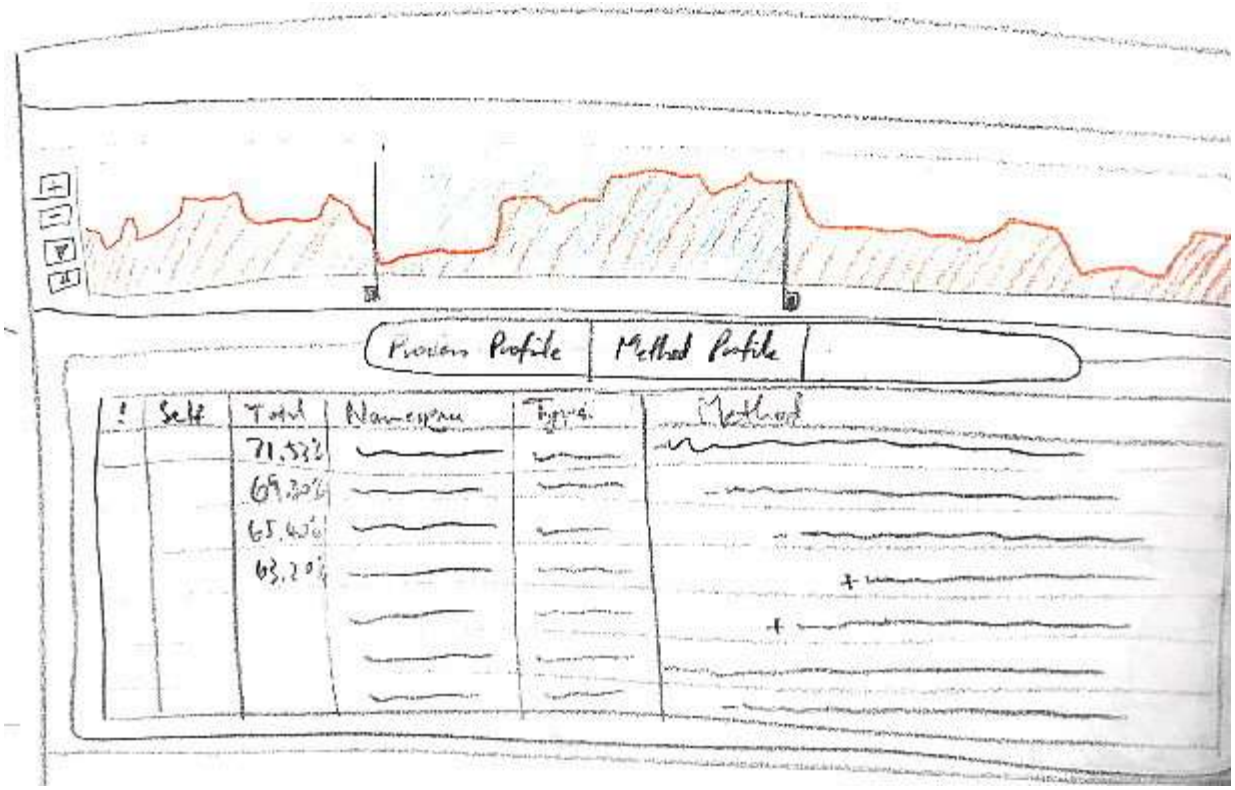
Improving Ease of Use



Improving Ease of Use

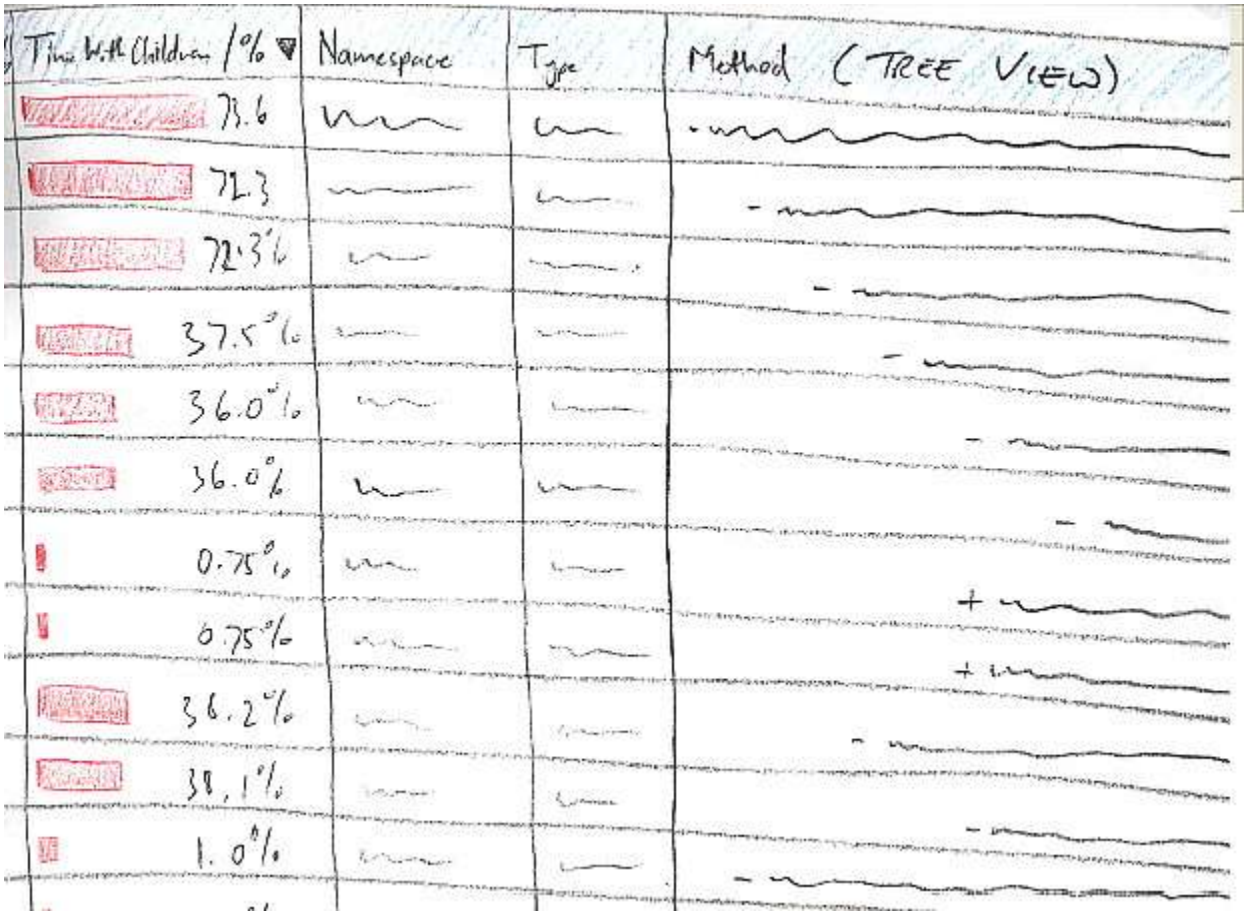


Improving Ease of Use

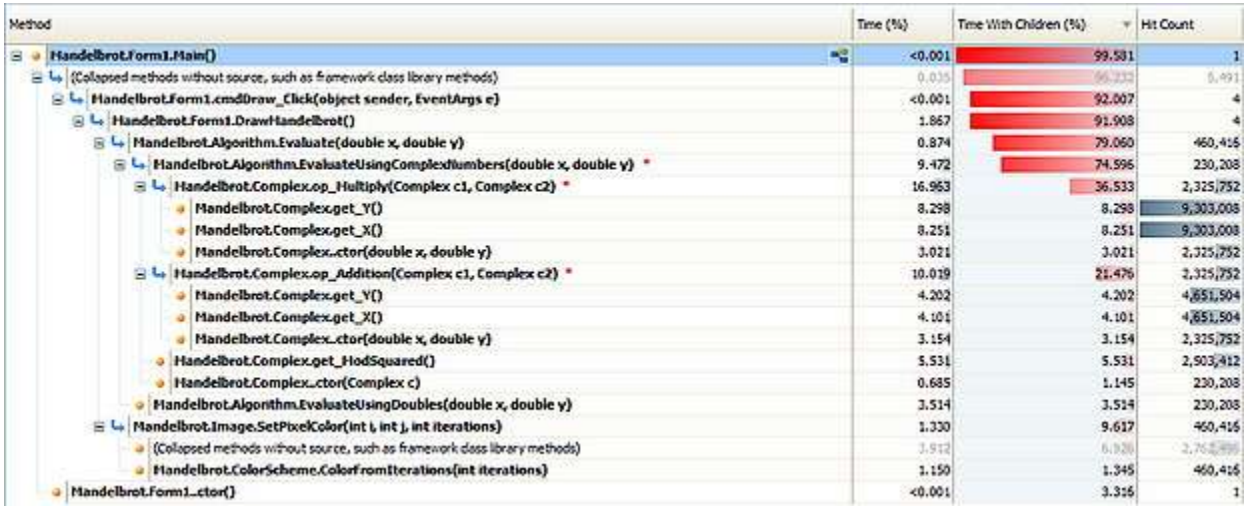


Improving Ease of Use

Improving Ease of Use

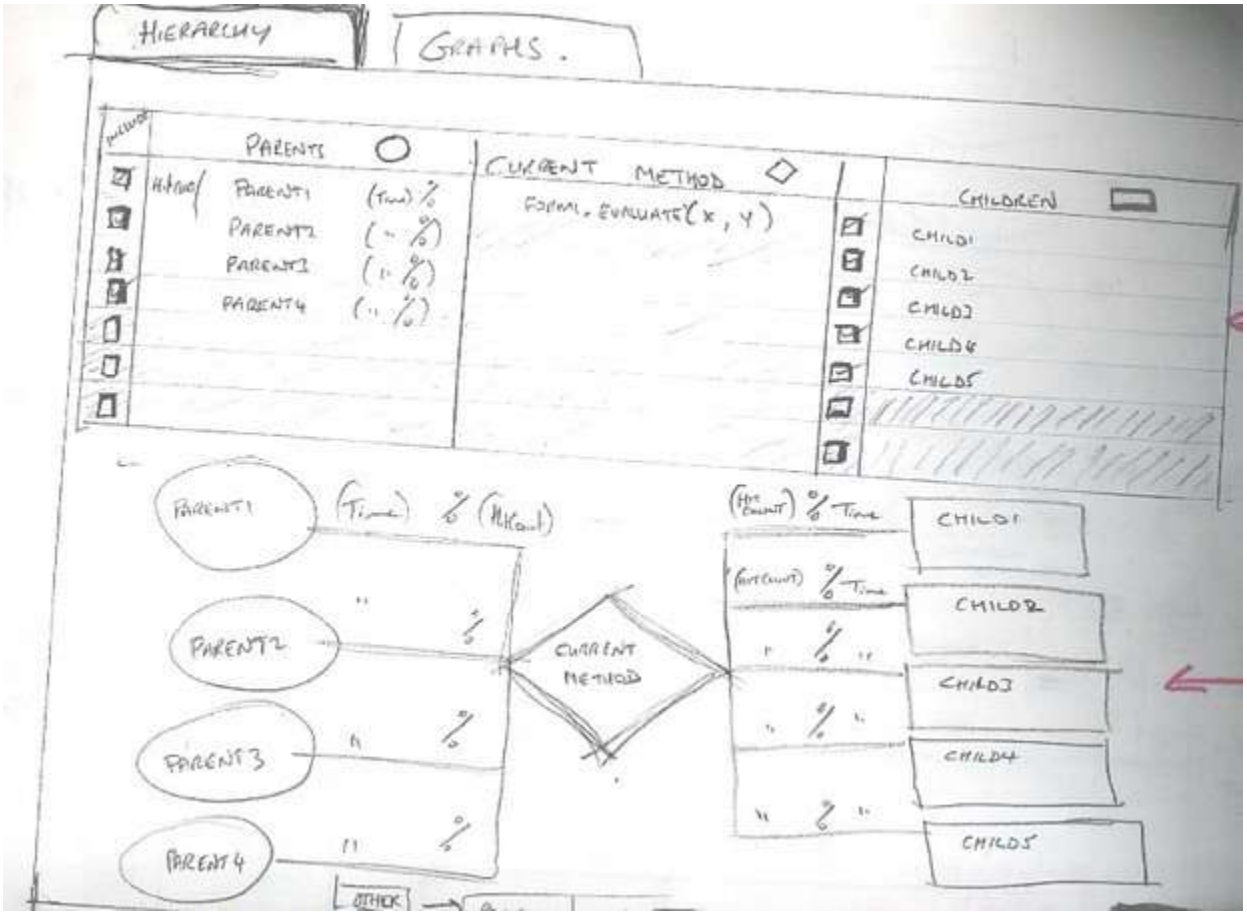


Improving Ease of Use

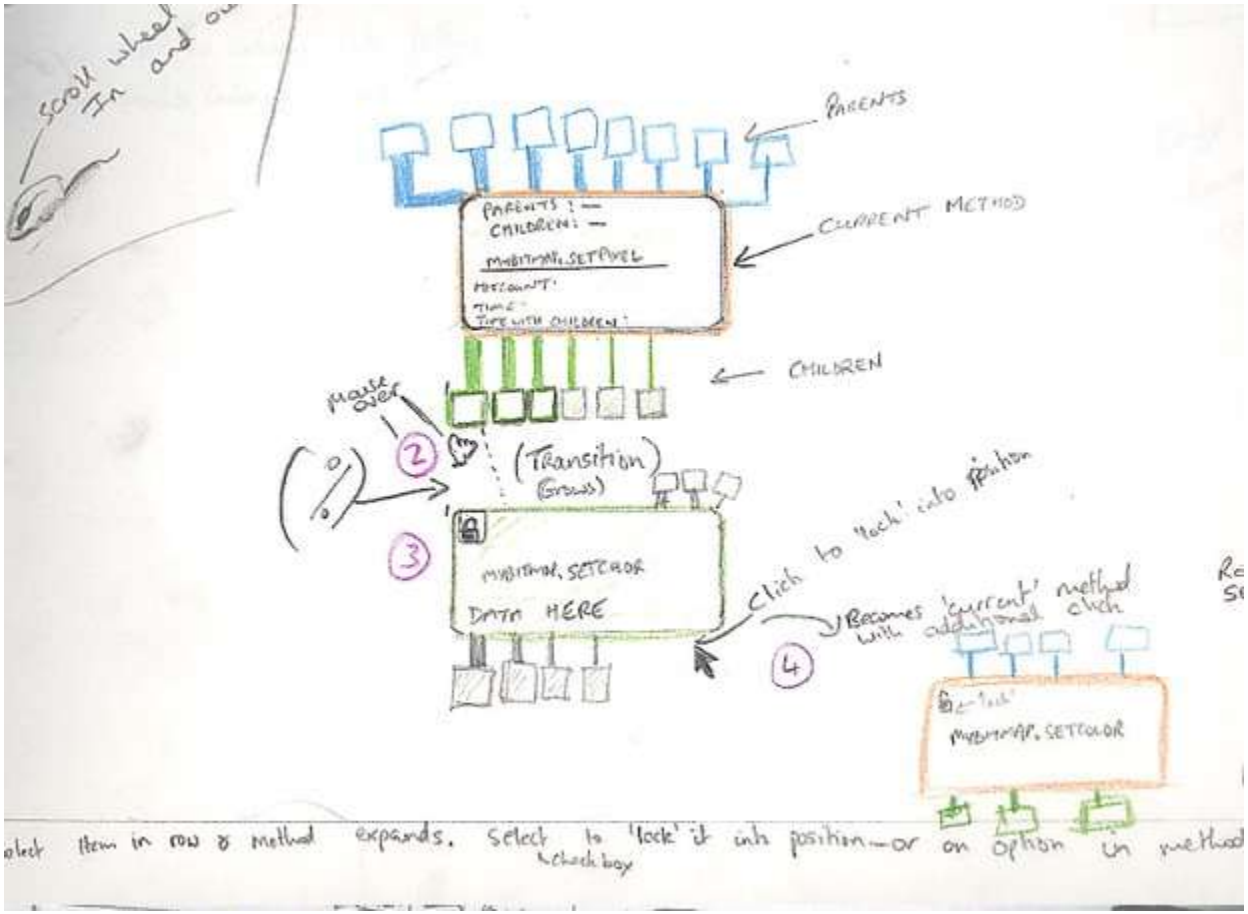


Method	Time (%)	Time With Children (%)	Hit Count
Mandelbrot.Form1.Main()	<0.001	99.581	1
(Collapsed methods without source, such as framework class library methods)	0.035	86.232	5,491
Mandelbrot.Form1.cmdDraw_Click(object sender, EventArgs e)	<0.001	92.007	4
Mandelbrot.Form1.DrawMandelbrot()	1.867	91.908	4
Mandelbrot.Algorithm.Evaluate(double x, double y)	0.874	79.060	460,416
Mandelbrot.Algorithm.EvaluateUsingComplexNumbers(double x, double y) *	9.472	74.596	230,208
Mandelbrot.Complex.op_Multiply(Complex c1, Complex c2) *	16.963	36.533	2,325,752
Mandelbrot.Complex.get_Y()	8.298	8.298	9,303,008
Mandelbrot.Complex.get_X()	8.251	8.251	9,303,008
Mandelbrot.Complex..ctor(double x, double y)	3.021	3.021	2,325,752
Mandelbrot.Complex.op_Addition(Complex c1, Complex c2) *	10.019	21.476	2,325,752
Mandelbrot.Complex.get_Y()	4.202	4.202	4,651,504
Mandelbrot.Complex.get_X()	4.101	4.101	4,651,504
Mandelbrot.Complex..ctor(double x, double y)	3.154	3.154	2,325,752
Mandelbrot.Complex.get_ModSquared()	5.531	5.531	2,903,412
Mandelbrot.Complex..ctor(Complex c)	0.685	1.145	230,208
Mandelbrot.Algorithm.EvaluateUsingDoubles(double x, double y)	3.514	3.514	230,208
Mandelbrot.Image.SetPixelColor(int i, int j, int iterations)	1.330	9.617	460,416
(Collapsed methods without source, such as framework class library methods)	3.912	6.928	2,764,496
Mandelbrot.ColorScheme.ColorFromIterations(int iterations)	1.150	1.345	460,416
Mandelbrot.Form1..ctor()	<0.001	3.316	1

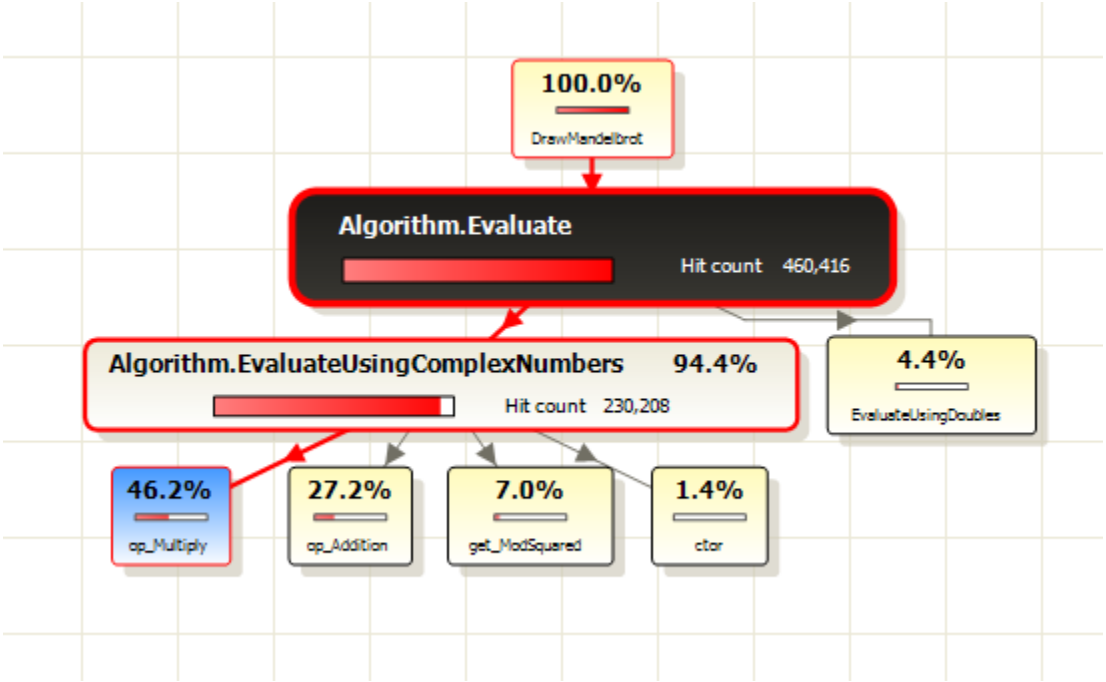
Improving Ease of Use



Improving Ease of Use



Improving Ease of Use



Running Usability Sessions

- Easy to do!
- Keep it cheap
- Explain the aims of the session to the user:
  - You are testing the software NOT the user
  - Don’t always answer their questions
    - Real users don’t have an expert sat next to them
- If they struggle for too long help them out with the specific issue
- If you know there is a problem help them sooner
- Remote sessions are fine

ANY QUESTIONS?