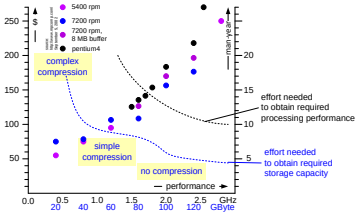
The realization view

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

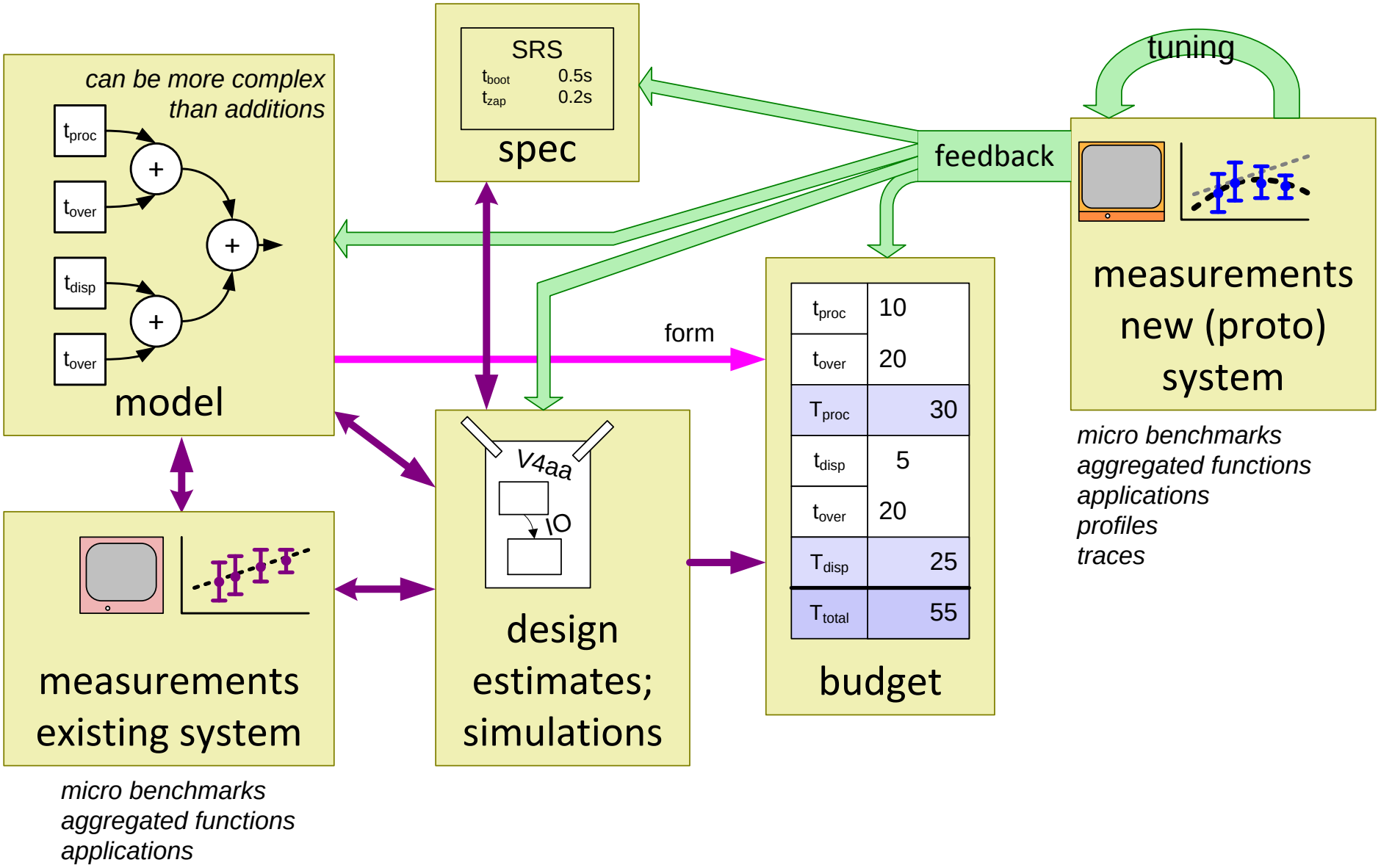
Abstract

The realization view looks at the actual technologies used and the actual implementation. Methods used here are logarithmic views, micro-benchmarks and budgets.
Analysis methods with respect to safety, reliability and security provide a link back to the functional and conceptual views.

August 21, 2020
status: preliminary draft
version: 0.1



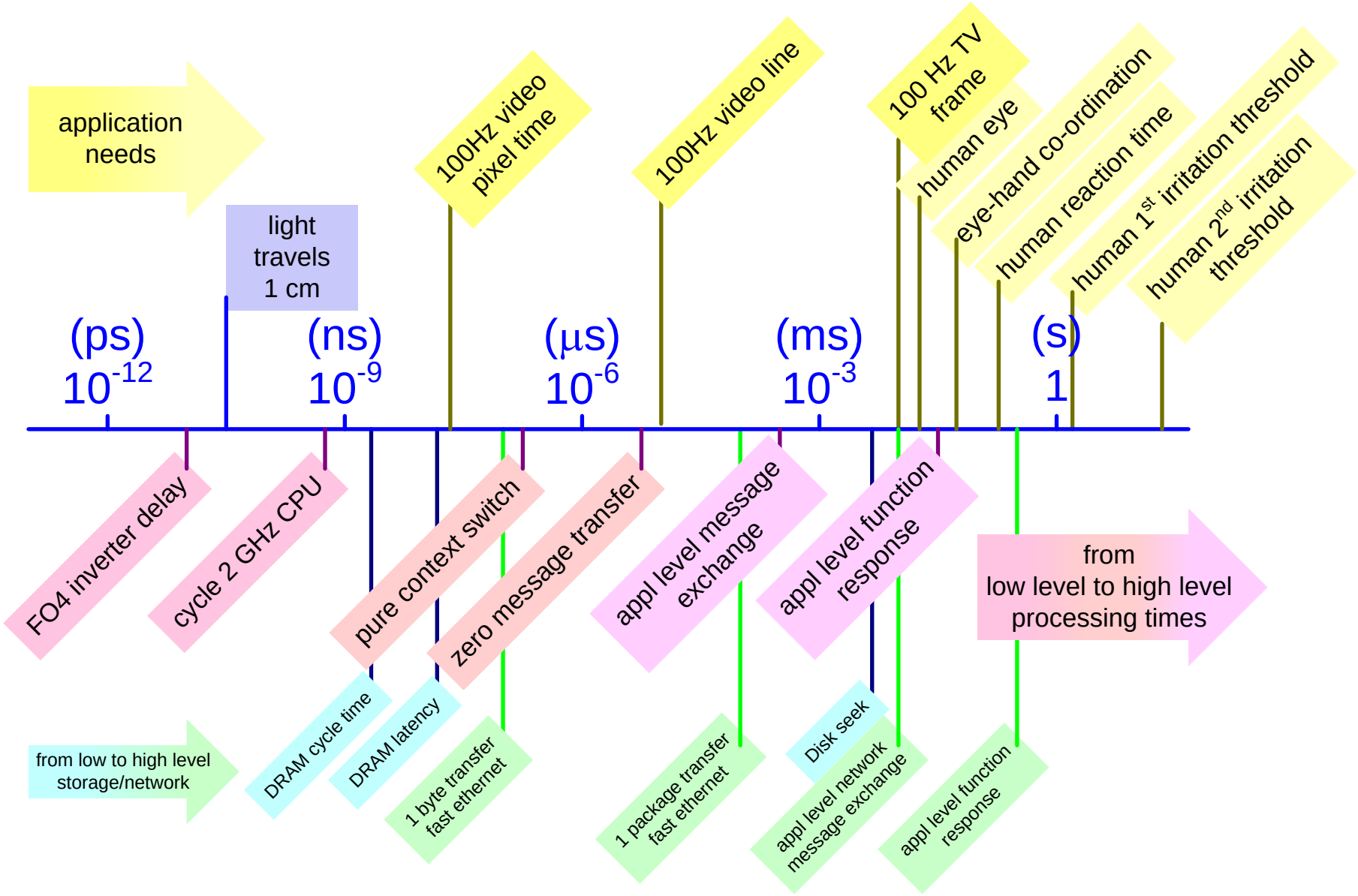
Budget based design flow



Example of a memory budget

memory budget in Mbytes	code	obj data	bulk data	total
shared code	11.0			11.0
User Interface process	0.3	3.0	12.0	15.3
database server	0.3	3.2	3.0	6.5
print server	0.3	1.2	9.0	10.5
optical storage server	0.3	2.0	1.0	3.3
communication server	0.3	2.0	4.0	6.3
UNIX commands	0.3	0.2	0	0.5
compute server	0.3	0.5	6.0	6.8
system monitor	0.3	0.5	0	0.8
application SW total	13.4	12.6	35.0	61.0
UNIX Solaris 2.x				10.0
file cache				3.0
total				74.0

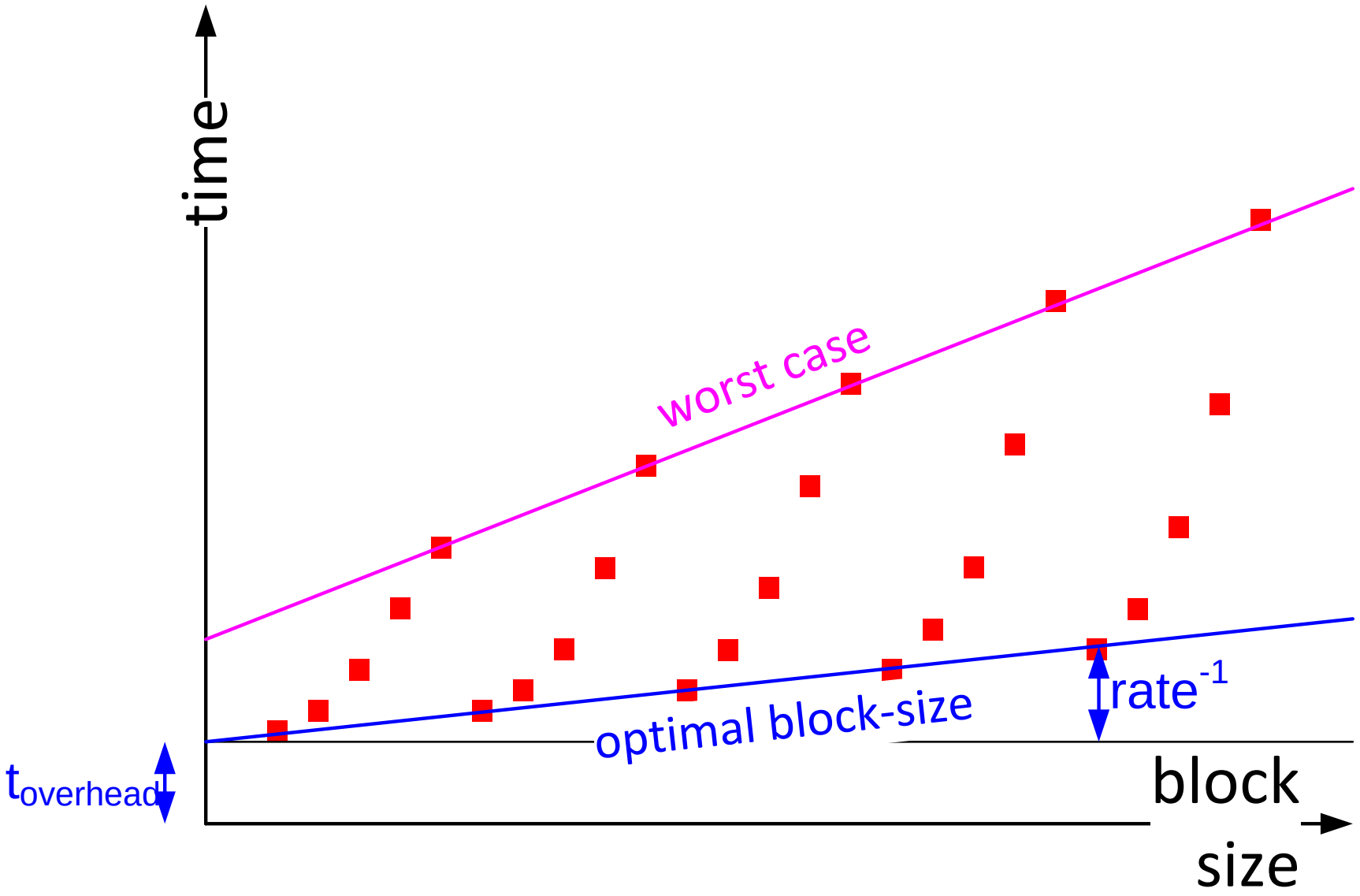
Actual timing on logarithmic scale



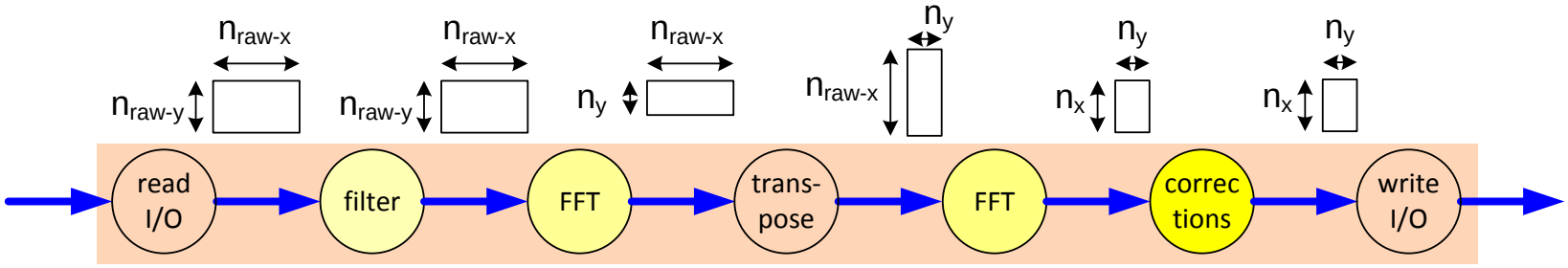
Typical micro benchmarks for timing aspects

	infrequent operations, often time-intensive	*often repeated operations*
database	start session finish session	perform transaction query
network, I/O	open connection close connection	transfer data
high level construction	component creation component destruction	method invocation same scope other context
low level construction	object creation object destruction	method invocation
basic programming	memory allocation memory free	function call loop overhead basic operations (add, mul, load, store)
OS	task, thread creation	task switch interrupt response
HW	power up, power down boot	cache flush low level data transfer

The transfer time as function of blocksize

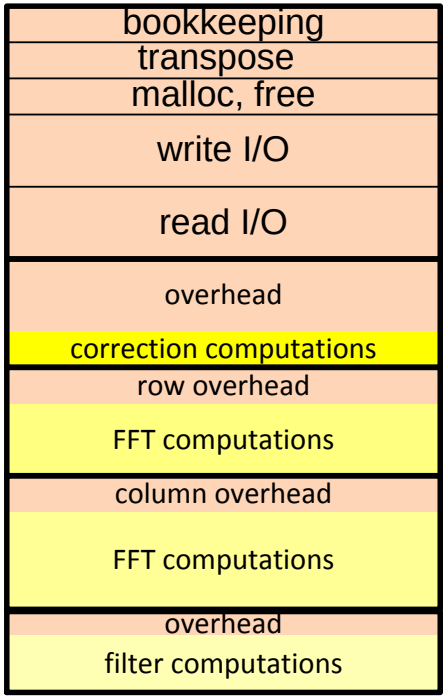


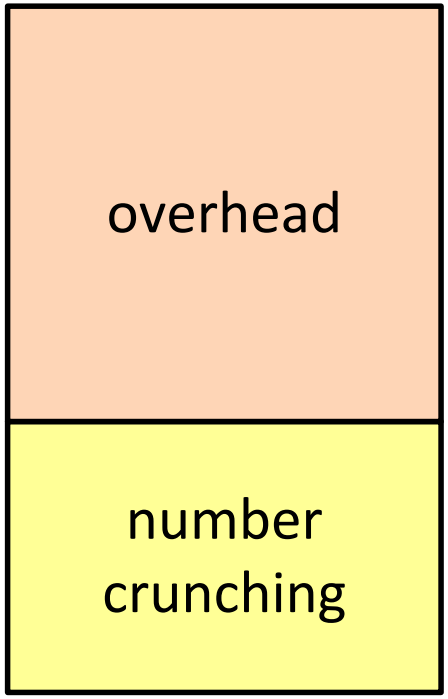
Performance evaluation



$$t_{recon} = t_{filter}(n_{raw-x}, n_{raw-y}) + n_{raw-x} * (t_{fft}(n_{raw-y}) + t_{col-overhead}) + n_y * (t_{fft}(n_{raw-x}) + t_{row-overhead}) + t_{corrections}(n_x, n_y) + t_{read\ I/O} + t_{transpose} + t_{write\ I/O} + t_{control-overhead}$$

$$t_{fft}(n) = c_{fft} * n * \log(n)$$





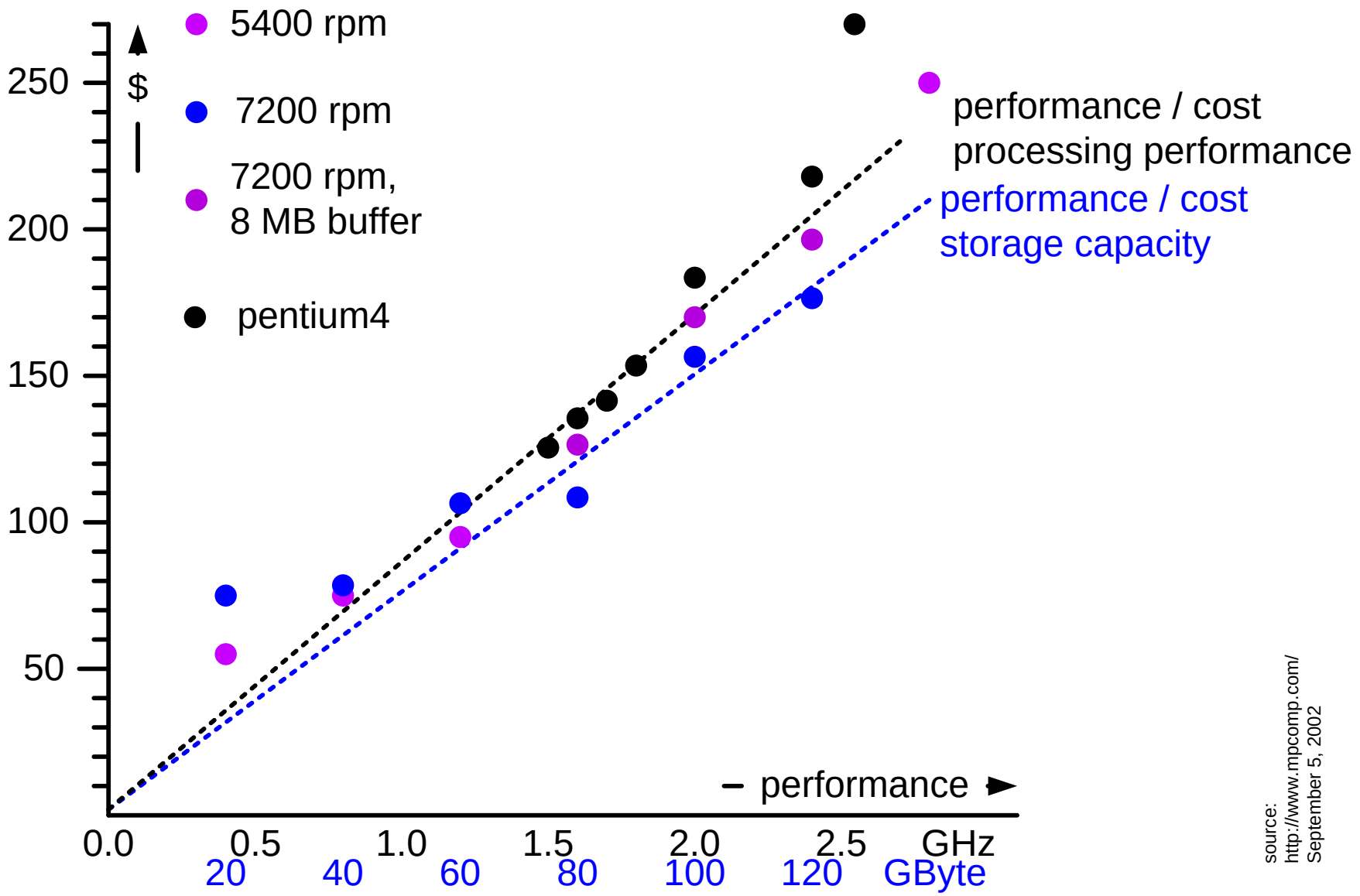
focus on overhead reduction

is more important

than faster algorithms

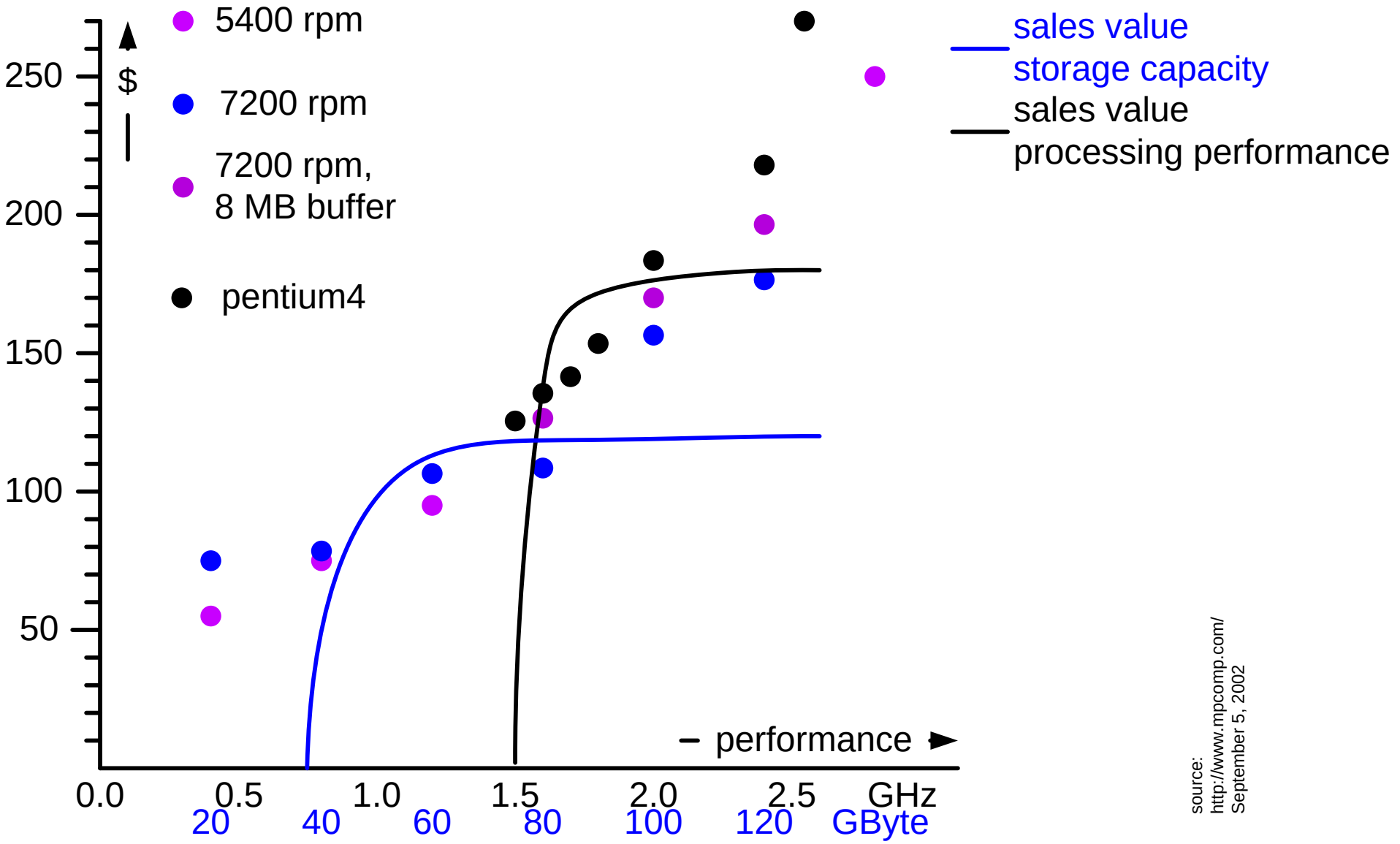
this is not an excuse for sloppy algorithms

Performance Cost, input data



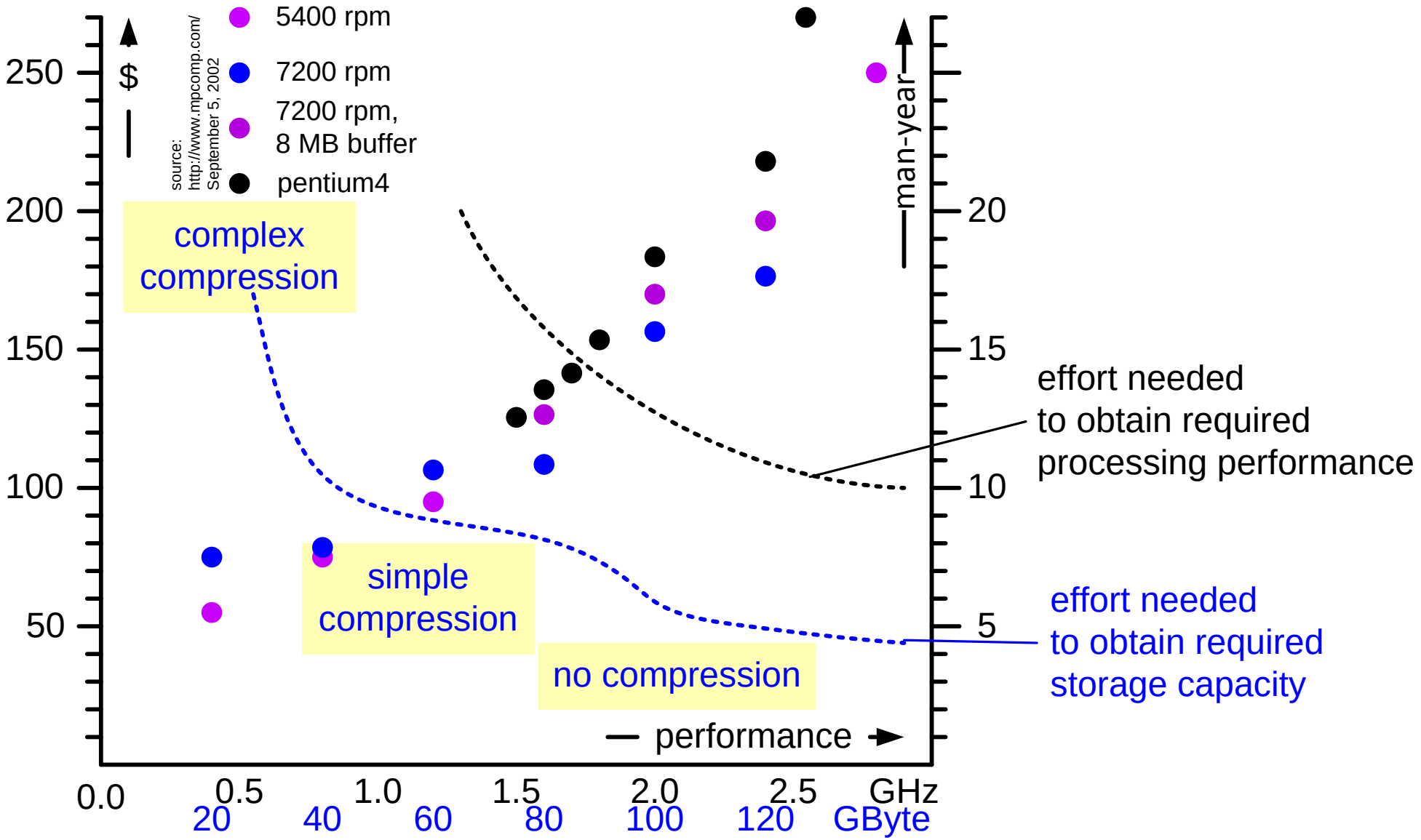
source:
http://www.mpcomp.com/
September 5, 2002

Performance Cost, choice based on sales value

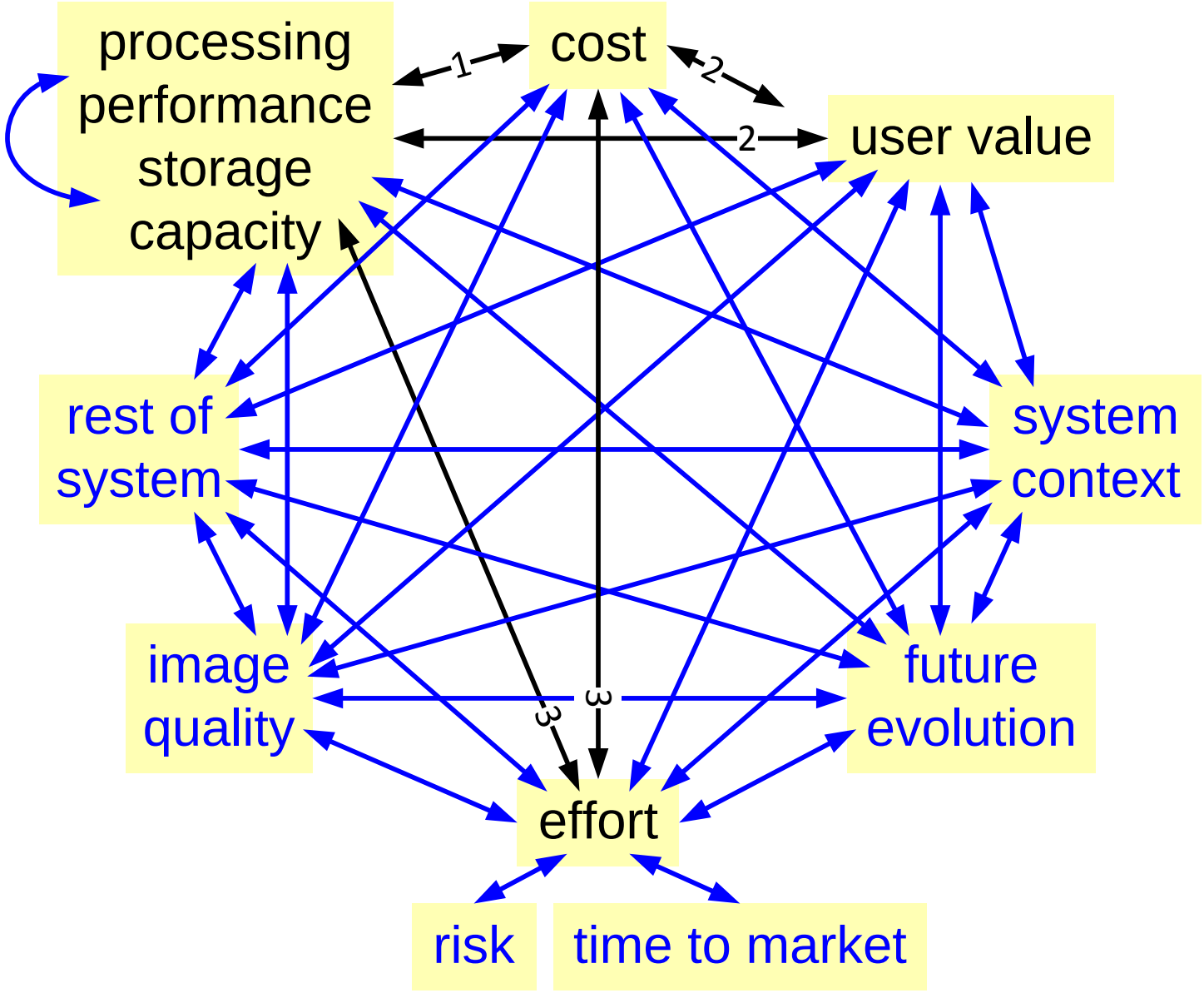


source:
http://www.mpcomp.com/
September 5, 2002

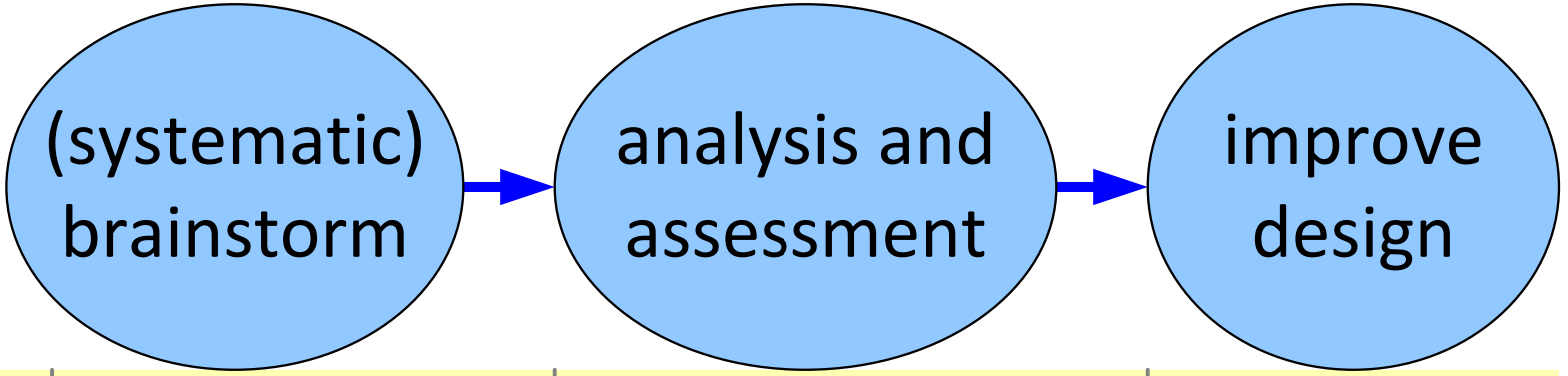
Performance Cost, effort consequences



But many many other considerations



Safety, Reliability and Security analysis methods



safety hazard analysis	potential hazards	probability severity	measures
reliability FMEA	failure modes	effects	measures
security	vulnerability risks	consequences	measures