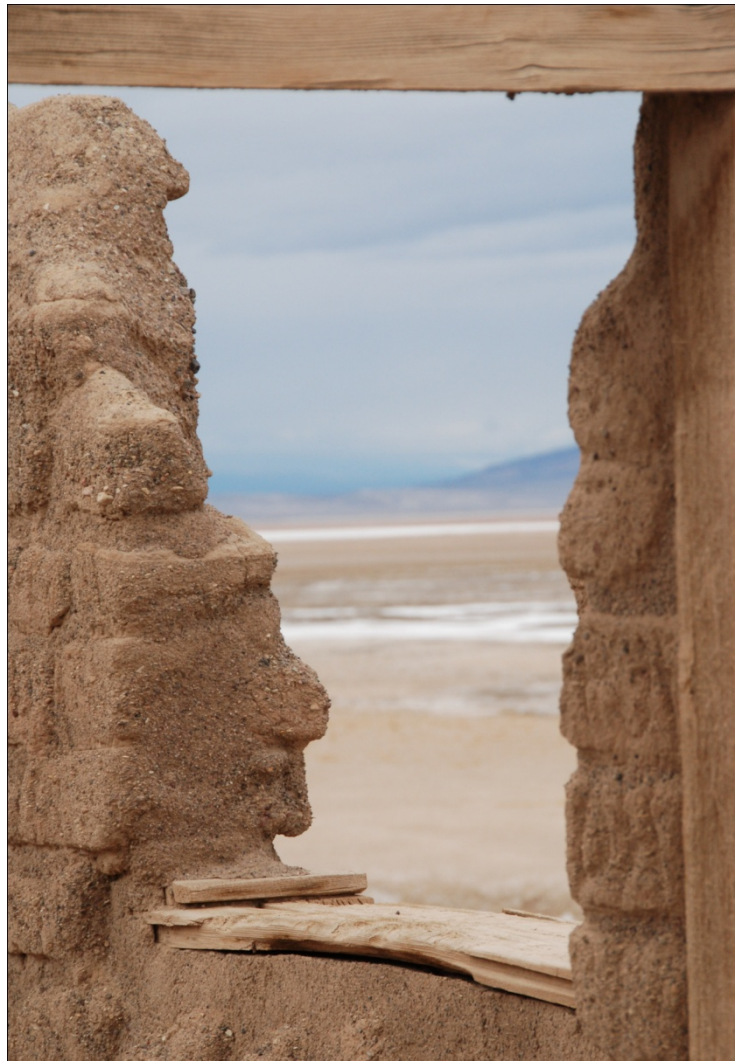
Is There Magic Associated with Software Benchmarks?

Reifer Consultants LLC
14820 N. Dragons Breath Lane
Prescott, AZ 86305-5644

Purpose of Talk

- **Aim** – summarize over three decades of experience generating software cost, quality and productivity benchmarks
- **Agenda**
  - Set the stage
  - Definitions
  - Discuss twelve major lessons learned
  - Conclusions



Setting the Stage

- Competition is ever-present
- Cost, quality and productivity are always an issue
- New approaches are continuously emerging
- New technologies always up and coming
- Improvements always desired
- Which way to do – always a question
- Cycle repeats itself

On Benchmarking

Benchmarks

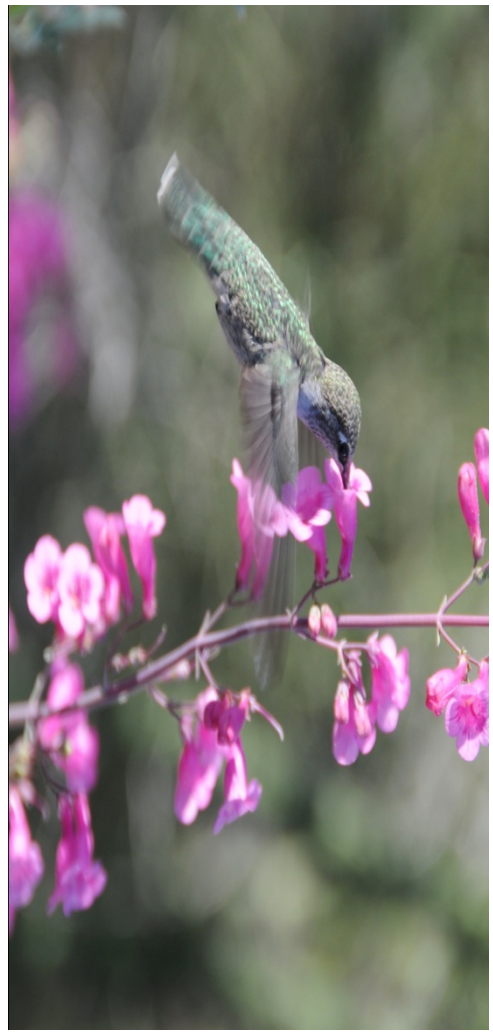
- Set "*baselines*" for making intelligent comparisons
  - Within like organizations, firms, application domains and industries
- Use "hard" numbers to confirm perceptions, traditions and trends

Common Misconceptions

- Productivity, cost and quality baselines do not exist
  - Every organization has folklore, others have numbers based on solid data
- Cost = Price
  - G&A, profit and variations in cost accounting practices can add to price
- Quality is hard to quantify
  - Both process and product measures exist and are used as yardsticks
- Productivity, cost and quality has not improved
  - While they have, it may be to your advantage to assume they have not

Types of Benchmarks

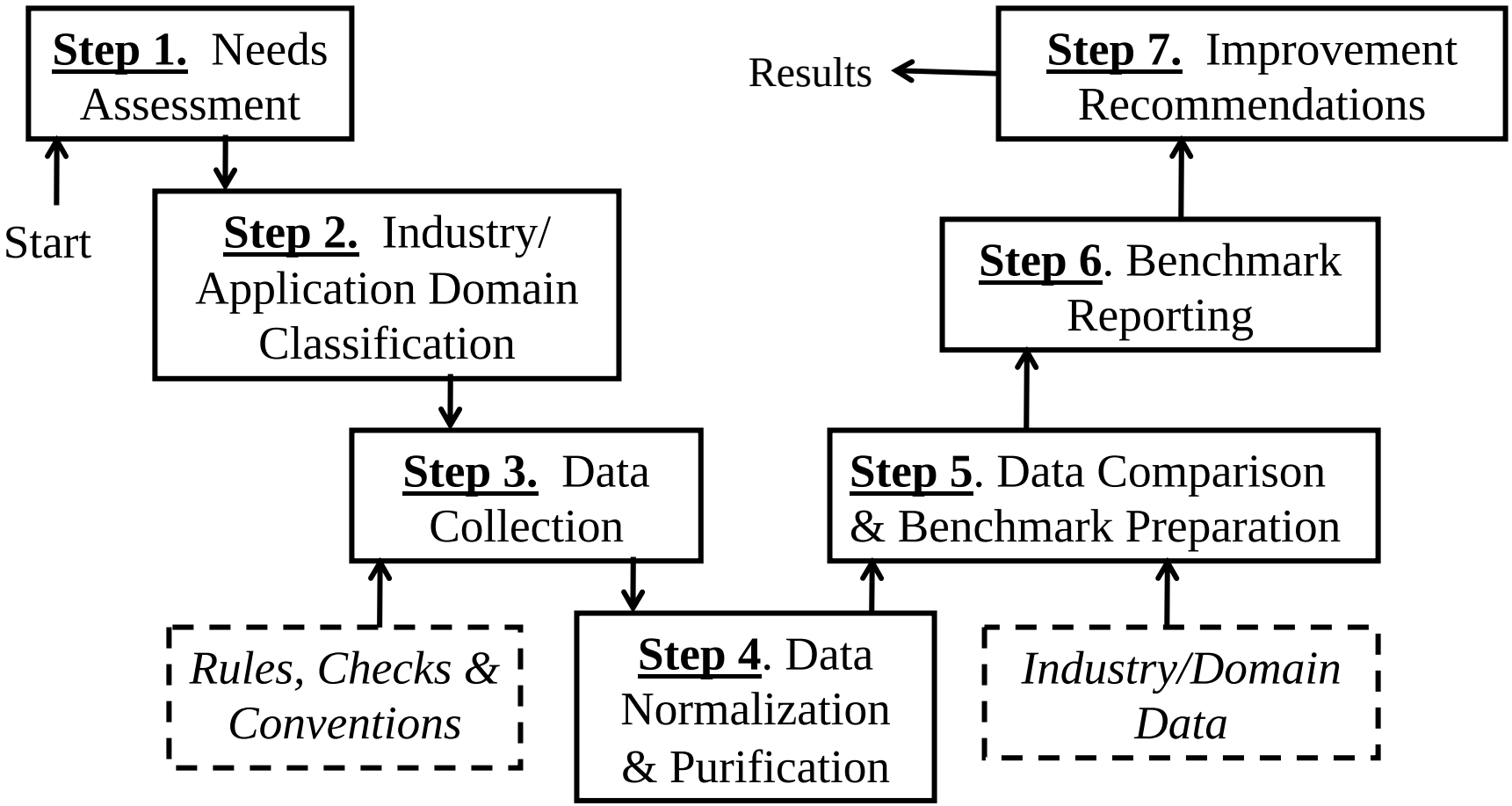
- Productivity and cost
- Quality and defect removal efficiency
- Customer satisfaction benchmarks
- Process assessments (CMMI, etc.)
- Staff turnover rates
- Staff compensation and benefits
- In-house vs. out-house performance
- Subcontractor performance
- Spending benchmarks
- Many others



Rules of Benchmarking

- Conceal the names of projects and participants
- Support only one set of core metrics (e.g., SLOC versus function points)
- Show overall data without amplifying details
- Conform to a well-defined scope and set of definitions
- Omit “soft factors’ and explain why projects vary
- Explain “how much” projects vary via ranges (three sigma if possible)
- Lead to improvement in software results

Benchmarking Process



Lessons Learned

- **Lesson 1** - Domain–specific benchmarking results are preferred by clients because they tend to be more accurate.
- **Lesson 2** - It is worth the effort to check the data as it is being submitted. Your results will be statistically better because the data will it be more coherent, complete and consistent.
- **Lesson 3** - Normalizing and purifying the data that you have collected takes a great deal of time and effort because you have to make sure that it hangs together statistically and makes sense from an engineering point-of-view.

Benchmarks Can Create Realistic Expectations For Productivity

Application Domain	No. Projects	Size Range (KESLOC)	Avg. Prod. (ESLOC/SM)	Range (ESLOC/SM)	Example Application
Automation	66	45 to 325	275	138 to 415	Factory automation
Command & Control	49	35 to 3,860	245	125 to 388	Command centers
IT-General	38	30 to 2,180	385	165 to 525	Payroll
Medical	35	45 to 985	305	152 to 453	Pharmacy
Military -All	**217**	**15 to 5,200**	**155**	**45 to 475**	**Seven subcategories**
Process Control	51	25 to 685	265	128 to 395	Factory automation
Software Tools	99	20 to 1,595	320	155 to 457	Debuggers/CM systems
Telecommunications	**61**	**25 to 2,950**	**300**	**145 to 465**	**Six subcategories**
Test Systems	45	28 to 575	310	152 to 477	Test equipment, etc.
Training/Simulators	33	45 to 1,750	295	137 to 442	Virtual reality simulator
Web Business	121	20 to 455	255	165 to 585	Client/server sites
TOTALS	**815**	**15 to 5,200**	**273**	**45 to 585**	

ESLOC used as common basis for comparing productivity

Especially when Domain-Specific

Military Domains	No. of Projects	Productivity	Telecomm. Domains	No. of Projects	Productivity
Airborne	41	48 to 185	Broadband	11	212 to 350
Ground	43	95 to 312	IPTV	7	195 to 275
Information Technology	31	143 to 475	Mobile	15	176 to 332
Medical	33	135 to 410	Satellite	11	145 to 228
Missile	25	45 to 146	Switching	8	162 to 273
Space	26	90 to 324	Test & Support	9	221 to 465
Trainers	18	155 to 428			
TOTALS	**217**	**45 to 475**	**TOTALS**	**61**	**145 to 465**

Notes

- Projects in each of these subcategories had many similar characteristics
- Telecom domains represent a new taxonomy developed in 2008 for large client
- In the past, we decomposed telecom projects into as many as 48 product categories
- Space includes both ground stations and on-board satellite systems

Lessons Learned (Continued)

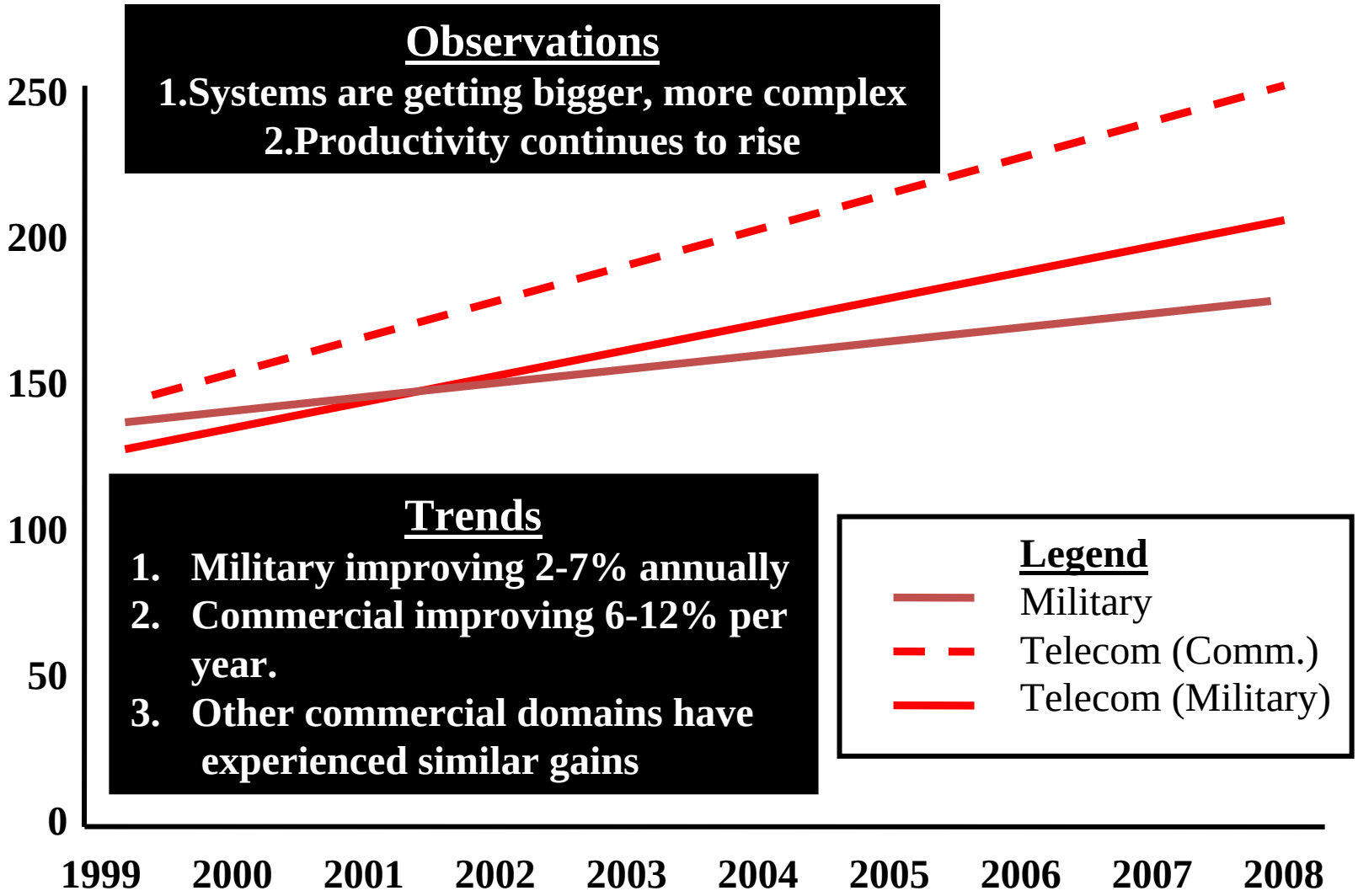
- **Lesson 4** – When data does not fall within expectations, look for the "root" cause. This search may reveal facts that might otherwise be overlooked.
- **Lesson 5** – Package information for different audiences using the same core data (effort in hours, size in SLOCs or function points, etc.). For example, balanced scorecards appeal to seniors because a lot of information can be communicated in just a few charts. Middle managers tend to like dashboards and graphics because they can visualize the details in a single glance.
- **Lesson 6** – When organizing benchmarks, let the data drive the taxonomy you use. Perform cluster analysis and statistical goodness of fit as your primary tools.

"Root Cause" Analysis

- Military – Missiles
  - In current benchmarks, both software cost and productivity have been increasing at same time
  - What is root cause?
    - **Excessive rework** – teams productively generating software to wrong requirements
    - **Lack of competition** – no controls on costs imposed by competitive market forces

- Military – Space
  - In current benchmarks, software costs have been rising at a higher rate than the norm during past four years
  - What is root cause?
    - **Growth** - Bigger systems built to tougher requirements
    - **Inflation** - higher costs for labor and materials
    - **Meddling** - excessive management oversight

Productivity Trends (by Domain)



Lessons Learned (Continued)

- **Lesson 7** – A good way to look at the data is by domains. Define these in terms of the applications your users relate to. In addition, use size as a secondary classification.
- **Lesson 8** – While not perfect, equivalent source lines of code (ESLOCs) are a useful size measure. However, to use them effectively, understand how they were derived.
- **Lesson 9** – When calculating cost or productivity, use the actual size of the software not an estimate.

Cost/SLOC

Application Domain	No. Projects	Avg. Cost ($/ESLOC)	Range ($/ESLOC)	Language Generation	Example Application
Automation	66	$42	$25 to $66	3GL	Factory automation
Command & Control+	49	$65	$30 to $121	4GL	Command centers
IT-General	38	$25	$18 to $55	All	Payroll
Medical	35	$33	$20 to $67	All	Pharmacy
Military -All	**217**	**$171**	**$100 to $350**	**All**	**Seven subcategories**
Process Control	51	$37	$25 to $62	3GL	Factory automation
Software Tools	99	$20	$14 to $48	4GL	Debuggers/CM systems
Telecommunications	**61**	**$50**	**$21 to $88**	**4GL**	**Six subcategories**
Test Systems	45	$49	$28 to $79	4GL	Test equipment, etc.
Training/Simulators	33	$55	$31 to $95	4GL	Virtual reality simulator
Web Business	121	$21	$12 to $38	5GL	Client/server sites
TOTALS	**815**	**$72**	**$12 to $350**	**All**	

3GL – Procedural languages **4GL – Object-oriented languages** **5GL – Natural languages**

<u>Myth</u> – it costs way less to out-source software overseas. On average, overseas cost in India and China is 60 to 75% less than in US. But, there is a 20% management cost and there are productivity penalties that need to be taken into account.

Lessons Learned (Continued)

- **Lesson 10** – When looking at benchmarks, understand exactly what the data means because you have to be careful when using the numbers.
- **Lesson 11** – Quality is king when looking at cost and productivity numbers. To truly understand the benchmarks, relate your numbers to quality norms and rank them accordingly.
- **Lesson 12** – While there is no magic associated with benchmarks, they serve as a useful tool when actual performance can be compared against trusted numbers.

Eight Year Quality Trend

(Average number of defects/KESLOC/year – reported post delivery)

Application Domain	Number Projects	Defect Range (Defects/KESLOC)	Normative Rate (Defects/KESLOC)	Notes
Automation	66	2 to 9	4	Factory automation
Command & Control	49	0.5 to 5	0.7	Command centers
IT-General	38	1 to 7	4	Payroll
Medical	35	1 to 5	2.5	Pharmacy
Military –All	**217**	**0.29 to 4.5**	**< 1.0**	**Seven subcategories**
Process Control	51	0.9 to 5	2	Factory automation
Software Tools	99	3 to 11	5.5	Debuggers, CM systems
Telecommunications	**61**	**1.2 to 7**	**3**	**Six subcategories**
Test Systems	45	1.5 to 9	3.5	Test equipment, etc.
Training/Simulation	33	2.1 to 8	3.9	Virtual reality simulator
Web Business	121	3 to 18	7	Client/server sites
TOTALS	**815**	**0.29 to 18**	**3.2**	

Military software has high quality because people's lives are at stake

Assumptions

- Defect rates are captured for the most part post delivery
- Defects are documented faults resulting in issues or actions
  - **Military** – trace issue to spec and track action to ensure closure occurs to agreed schedule/budget
  - **Commercial** – prioritize issues and resolve them on LOE basis; track closures; budget is LOE
- Productivity normalized using defect rates in advanced cultures
  - Easy to increase productivity at the expense of quality

Sources of Variation in Defects

- Test practices
  - Commercial uses independent test organizations and test-first methods, defense waits until later in life cycle to consider testing
- Development practices
  - Commercial views requirements as a learning rather than specification activity
  - Commercial develops frequent rapid prototypes, defense does inspections and develops paper rather than working products
  - Commercial has embraced less rigorous test processes, defense has not (i.e., they focus on capabilities rather than detailed specifications)
  - Both have embraced statistical process controls and Six Sigma
- Maintenance practices
  - Commercial has developers maintain their products, defense does not
  - Commercial builds regression test baselines, defense often does not
  - Commercial embraces beta testing involving the user during development, defense only does it in a limited manner

In Conclusion

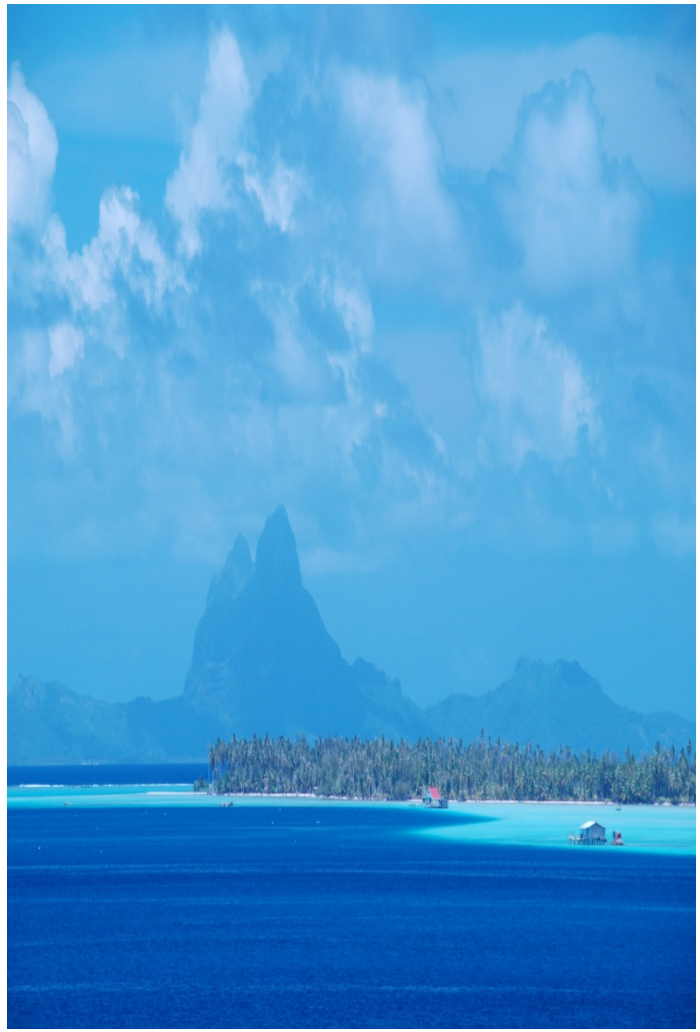
- Numbers can be your friend
  - Help you win battles and get your boss off your back
- When using the numbers
  - Make sure they are easy to collect and solid
  - Spin them to your advantage
  - Ready your organizations to make needed changes
- When trying to get changes approved
  - Align your numbers with your business goals
  - Use value engineering concepts and business cases to help others understand what the numbers mean
  - Use both business and technical justification to win your battles and the war of the budget

Making Improvements

- Benchmarks help you improve by helping you to:
  - Do the right things right the first time
  - Know what the right things to do are
  - Know why your cost, productivity and quality are varying
  - Focus on the 20% that provides the 80% yields (value-based)
  - Adopt those industry best practices that make sense for your organization

In Summary

- **Aim** – summarize over three decades of experience generating software cost, quality and productivity benchmarks
- **Final Lesson Learned (13)**
  - Trust benchmarks when arguing for change because the picture that they paint relative to performance is rarely wrong



Questions Or Comments

Donald J. Reifer

dreifer@earthlink.net

Phone: (928) 237-9060

When eating an elephant take one bite at a time.

Creighton Adams

Its time for IBM to perform and then talk, instead of talk and then perform

Louis Gerstner Jr., CEO IBM