Software Cost/Productivity/Quality: Have We Made Any Progress in the Past Decade?

Donald J. Reifer
Reifer Consultants, Inc.
P.O. Box 4046
Torrance, CA 90510
Phone: (310) 530-4493

Introduction

- Report results of an analysis of cost, productivity and quality trends during the past decade
  - Compare commercial and aerospace industry trends
  - Highlight the sources of variation
- Look at sources of variation and associated best practices
  - Pinpoint what is driving the trends
- Provide you insight into what you can do to take advantage of our analysis of trends

Setting the Stage

Within Aerospace/Defense:

- Rapid move to net-centric systems-of-systems
  - Future Combat System, GIG, Undernet, etc.
- Emphasis on joint, interoperable capabilities
  - Alliances involved as well as coalition partners
- More reliance on the contractor to lead efforts (LSI)
- A great deal of talk about new acquisition paradigms
- Lots of volatility in requirements and architecture
- Continued emphasis on process paradigms (CMMI)
- Increased use of COTS and open source products
- Focus continues to be on controlling costs and providing the warfighter with the edge with high technology

Setting the Stage

Commercial Firms:

- Continued out-sourcing, in-sourcing and globalization
  - The world is definitely flat with China and India taking command of IT worldwide
- Emphasis on enterprise-solutions that are net-ready and increasingly interoperable
- Increased focus on agile rather than disciplined methods
  - View CMMI as overkill and inappropriate for international teams
- Short-term considerations dominate as the US continues to lose its strategic edge in systems and software engineering
- Primary focus continues to placed on being first to market with competitive, feature-rich products

Two Different Worlds

	Aerospace	Commercial
Driving force	Ability to support the warfighter	Ability to penetrate markets and make money
Mindset	Pessimistic and risk adverse	Optimistic and willing to take risk if rewards high
Primary focus	Customer satisfaction and cost control	Customer satisfaction and market dominance
Process used	Acquisition mindset with lots of oversight/discipline	R&D mindset with some oversight/ discipline
Driving issues	• Understanding what is really wanted • Bureaucracy and overkill	• Being nimble, but in control • Devising discriminators

Common Myths

Aerospace/Defense

- They rely on discipline (all projects use common processes)
- Quality is number “1”
- They are pushing the state-of-the-art
  - Have lots of legacy issues to address
- Teams are extremely skilled
- There is lot of waste and inefficiencies

Commercial

- They lack discipline (they hack products together)
  - Many are very disciplined
- Quality does not matter
- They are primarily concerned with legacy
  - Lots of new technology is involved
- Teams are not skilled
- They are extremely efficient
  - There is lots of bureaucracy

Benchmarking Provides Insights

Benchmarks

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

Common Misconceptions

- Productivity, cost and quality baselines do not exist
  - Every organization has folklore, some 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

Software Productivity/Application Domain (ESLOC/SM)

Application Domain	No. Projects	Size Range (KESLOC)	Avg. Prod. (ESLOC/SM)	Range (ESLOC/SM)	Example Application
Automation	58	25 to 785	275	118 to 445	Factory automation
Banking	112	55 to 1,000	282	155 to 550	Loan processing, etc.
C&C	55	35 to 4,500	250	95 to 350	Command centers
Data Processing	135	20 to 950	325	165 to 500	DB-intensive systems
Environment/Tools	75	15 to 7,500	400	143 to 758	CASE, compilers, etc.
Military -All	225	15 to 8,125	152	45 to 330	See subcategories
Scientific	45	28 to 790	215	110 to 450	Seismic processing
Telecom.	85	15 to 2,280	275	175 to 490	See subcategories
Test	65	20 to 800	220	100 to 485	Test equipment, etc.
Trainers/Simulations	30	150 to 1,200	255	143 to 830	Virtual reality simulator
Web Business	115	10 to 750	330	190 to 985	Client/server sites
TOTALS	**1,000**	**10 to 8,125**	**262**	**45 to 985**	

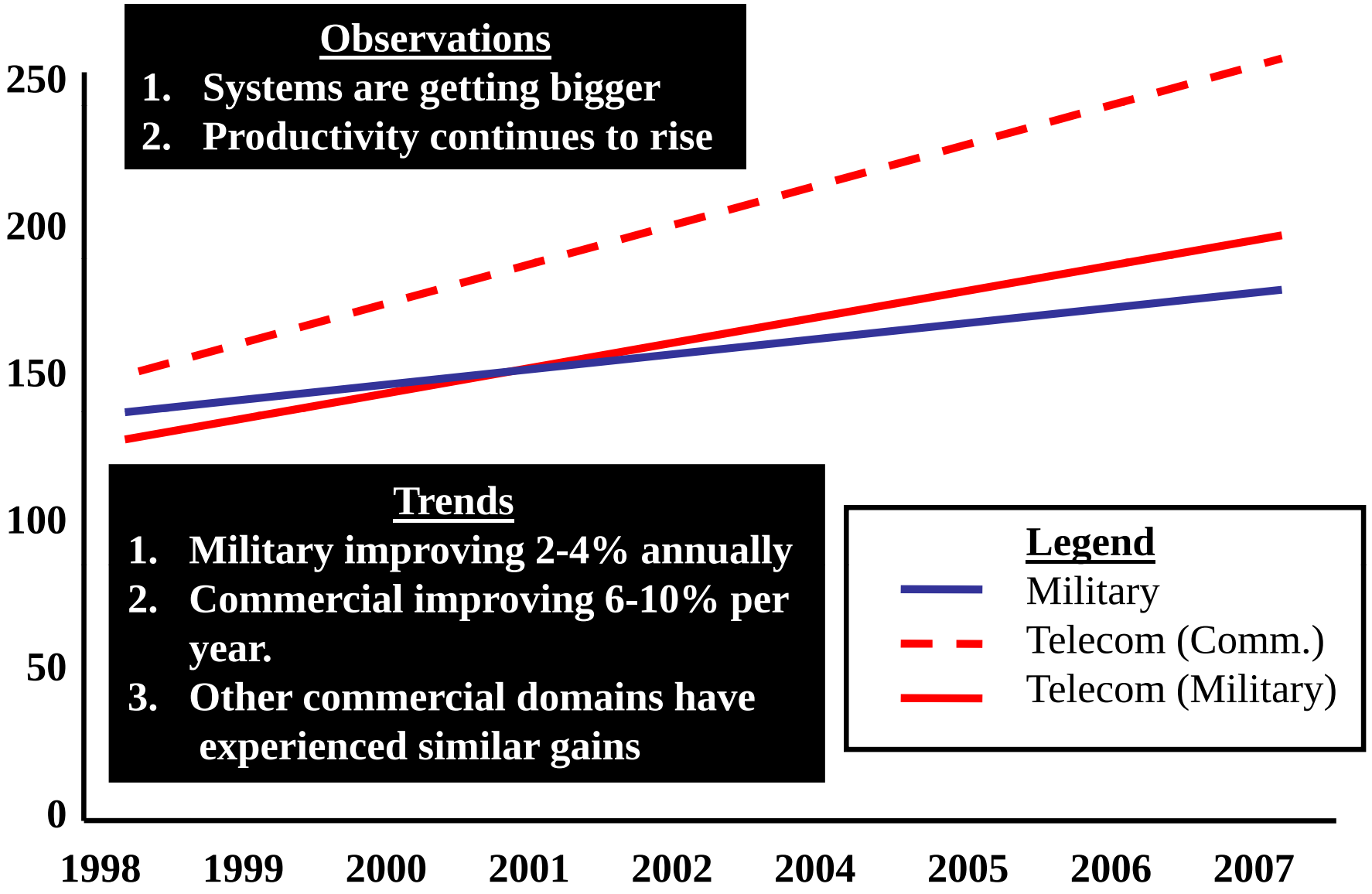
ESLOC used as common basis for comparing productivity because it was readily available

Assumptions

- 1,000 projects from our databases were completed within last ten years
- Scope of projects
  - Military
    - From SRR until handoff to the systems test bench
  - Commercial
    - Project start to customer sell-off
- Includes only directly charged labor hours
- Average hours/month = 152
- ESLOC per SEI definitions

- Function and feature points converted to SLOC using IFPUG and other schemes
- Different life cycles used were normalized around scope of standard project
- Agile data is included as it is consistent from end game point-of-view
- Foreign data was included only when project was international in scope
  - Some filtering was performed

Productivity Trends (by Domain)



Sources of Variation in Productivity

- Application domain
  - Data globs within like applications domains
  - Domains have similar characteristics including architectures
- People
  - Worker skills and experience vary greatly as does the extent to which high performers are rewarded (they continue to tend to be work/not salary driven)
  - Aerospace workforce is older and more experienced (age = 48 vs. 32 years and experience = 25+ vs. 10+ years)
  - US and foreign databases kept separate because data is non-homogeneous (not true for international projects)
- Amount of work to be done
  - There is a lot more non-productive work in defense

Time-to-Market is most often a constraint and not a variable on most projects

Software Productivity Best Practices

Ranking	Practices	Mapping
1	Learning organization; knowledge management infrastructure; university partnerships; staff projects based on the job's core competencies	Skilled and experienced workforce
2	Interdisciplinary teams; Team Software Process (TSP); facilitating infrastructure; co-location of test and integration team	Collaborative teams
3	Goal-directed reward structure; self-metric environment; value-based culture; self-directed teams; culture takes risks/celebrates mistakes	Empowered workforce
4	ISO gated process; CMMI; technology maturity levels (TRL); other frameworks (SPICE)	Process maturity
5	Statistical process control; black belts	Process efficiency
6	Environment hubs, integrated software engineering environments; regression test tools	Process automation

Cost/SLOC by Language/Domain

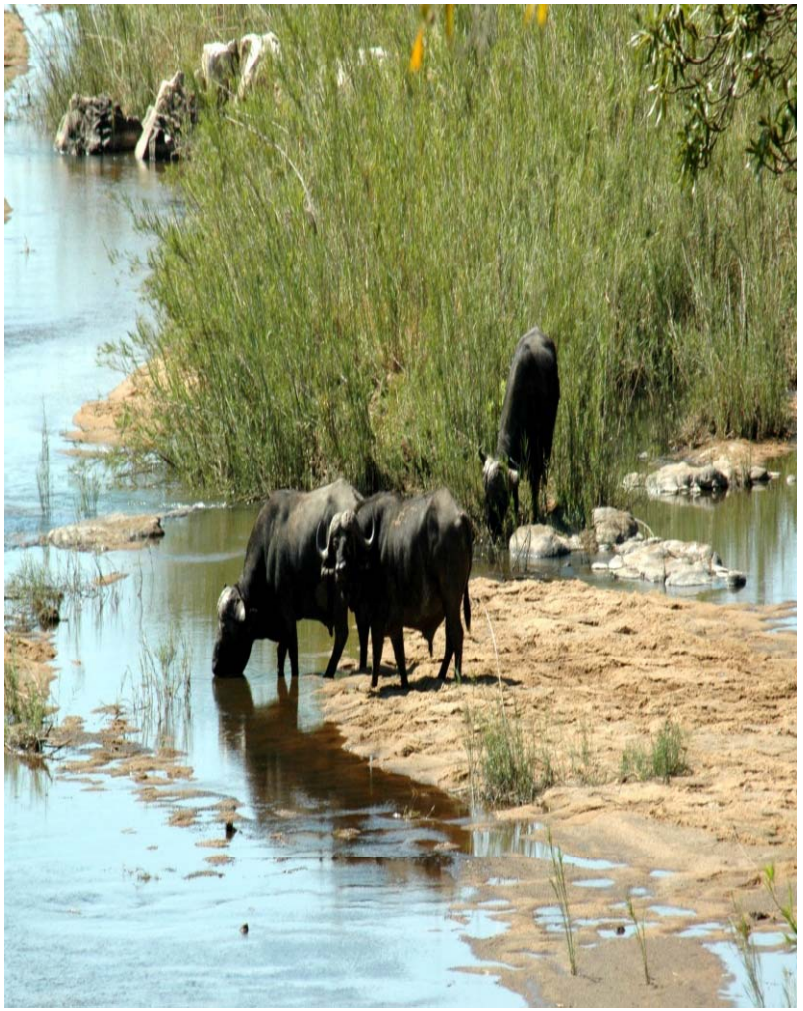
Application Domain	3GL	4GL	5GL	Norm	Notes
Automation+	50	35	20	25	Most implement ladder nets.
Banking+	40	25	15	30	Many moving to reuse, generators and templates.
Command & Control+	100	65	35	50	Movement to Java, reuse and COTS/Open Source
Data Processing+	40	25	15	20	Many have moved from COBOL to Java.
Environment/Tools+	40	30	20	25	Movement to Java, reuse and COTS/Open Source
Military -All	200	125	75	145	While many use 3G, many have moved to OO languages and real-time Java. Ada declining.
Scientific+	100	75	50	65	Problem remains hard – still use 3GL
Telecommunications++	75	45	25	40	Movement from C/C++ to Java and C#
Test+	50	25	20	35	Movement to Java, reuse and COTS/Open Source
Trainers/Simulations++	75	50	25	40	Movement to Java , reuse and COTS/Open Source
Web Business+	50	25	10	20	Most use Java, HTML, PERL, etc. and .NET

+ Primarily commercial projects **++ Mix of military and commercial projects**

Myth – it costs less to out-source software overseas. On average, productivity overseas in India and China is 10-20% less than in US and there is a 20% cost to manage sources

Assumptions

- U.S. dollars ($) used as basis of analysis
- Labor costs assumed were:
  - Aerospace/military (w/o profit)
    - Staff year = $200K or $16.6K/staff month
  - Commercial (w/o profit)
    - Staff year = $120K or $10K/staff month
- Many language used as norm
  - Java most common for commercial
  - C/C++ most common for military
  - .NET and 5GLs very popular for web applications
  - Ada use was almost non-existent



Cost Trends

- Cost and productivity are related but driven by different factors
  - If not careful, you can increase them both at the same time
- Costs have decreased by about 10 to 20% per year
  - Trend towards the use of 4GL and 5GL has simplified effort and allowed user to pick up some of the development
  - Trend towards the use of COTS and Open Source has reduced the quantity of work required to deliver capability
- Several counter-intuitive trends have emerged
  - Maintenance costs have not been reduced because costs to maintain COTS and Open Source are same as new software
  - Maintenance is driven by head count and not requirements

Sources of Variation in Cost

- Demographics of how the work is organized and costs are captured influence what the resulting cost/SLOC
  - Aerospace separates software development from maintenance work leading to extra costs
- Labor rates
  - Aerospace labor rates are higher because they bundle added expenses as part of their costs:
    - Capital costs of engineering (labs, test facilities, etc.)
    - Benefit plans and R&D costs are charged to overhead accounts
  - Commercial firms provide stock options to motivate troops rather than benefit plans that impact overhead
- Subcontracting, in-sourcing and outsourcing costs
  - Management and communications costs are often underestimated
- Use of COTS and Open Source
  - Costs of licensing and maintenance can be severely underestimated

Cost Best Practices

Ranking	Practices	Mapping
1	Good management; COTS, Open Source and legacy utilization; product measurement and metrics (linked to tracking defects and growth)	Reduced job scope and complexity
2	Visual environments; simulation-based acquisition; specification methods (UML-2, agile methods, performance threads, etc.); Kaizen	Do the right job right the first time
3	Architecture-based initiatives; facilitating infrastructure; modern reward systems	Systematic reuse
4	Subcontract work; separate cost centers; in-sourcing and outsourcing; support contractors	Reduced unit costs
5	Daily meeting; war rooms; daily builds; endgame management; retrospectives	Rework reduction
6	Orthogonal defect control; reliability models; metrics and modeling; endgame management	Defect reduction

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	58	2 to 10	5	Factory automation
Banking	112	3 to 12	6	Loan processing, ATM
Command & Control	55	0.5 to 5	1	Command centers
Data Processing	135	2 to 12	7	DB-intensive systems
Environment/Tools	75	5 to 15	8	CASE, compilers, etc.
Military –All	225	0.2 to 3	< 1.0	See subcategories
Scientific	45	0.9 to 5	2	Seismic processing
Telecommunications	85	3 to 12	5	See subcategories
Test	65	3 to 14	7	Test equipment, devices
Trainers/Simulations	230	2 to 13	5	Virtual reality simulator
Web Business	115	4 to 21	12	Client/server sites

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

Assumptions

- Defect rates are captured 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, defense waits until later in life cycle
  - Some agile project use test-first programming methodology
- Development practices
  - Commercial develops frequent rapid prototypes, defense does inspections and develops paper well in advance of code
  - 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 development team maintain their products, defense does not
  - Commercial builds regression test baselines, defense often does not
  - Commercial uses user beta testing throughout development, defense does not

Quality Best Practices

Ranking	Practices	Mapping
1	Quality as a teacher, not an inspector philosophy - Best people assigned to quality organization - In addition to ensuring process, incentives aimed at improving product are provided	Better teamwork and improved products
2	Independent test organization used - Gets test group involved early in the process - Scope of organization includes QA and CM	More effective roles and responsibilities
3	Value-based testing approach used to get the most bang for the buck	Testing effort prioritized
4	Models and metrics used to guide test conduct - Determine “when have tested enough?”	Testing effort quantified
5	Test-first process with frequent prototyping and incremental deliveries	Disciplined test process

What Practices Work, What Don't

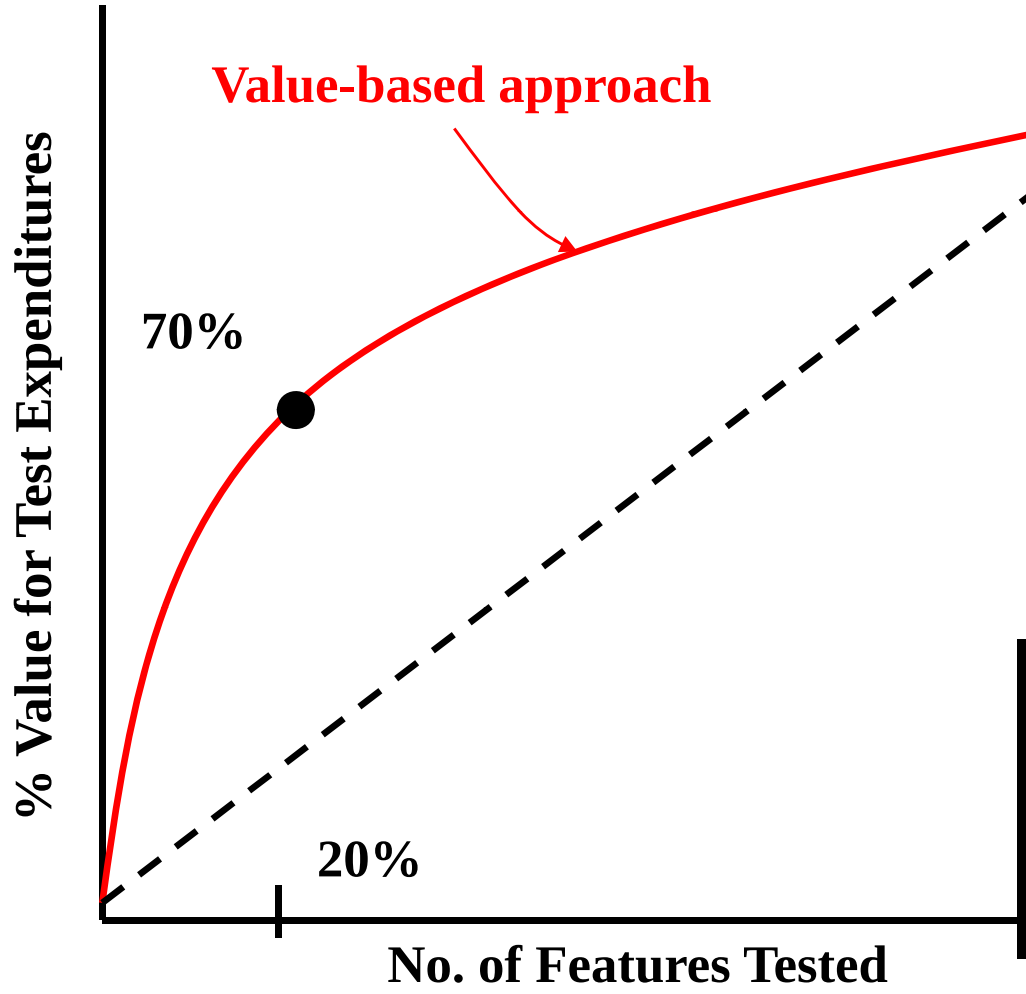
What Works

- Value-based testing
- Meaningful quality metrics and models
- Test-first process
- Inspections ***and*** (not ***or***) disciplined testing
- Static/dynamics analyzers
- Test models and metrics
- Regression test suite and baselines

What Doesn't Work

- Automated test generation
- Ad hoc testing
- Exhaustive testing (test until you're exhausted and then deliver to user)
- Staged testing phases
  - DT&E, IOT&E & OT&E
- IV&V (especially when vendor's aim is finding fault and they are not qualified)

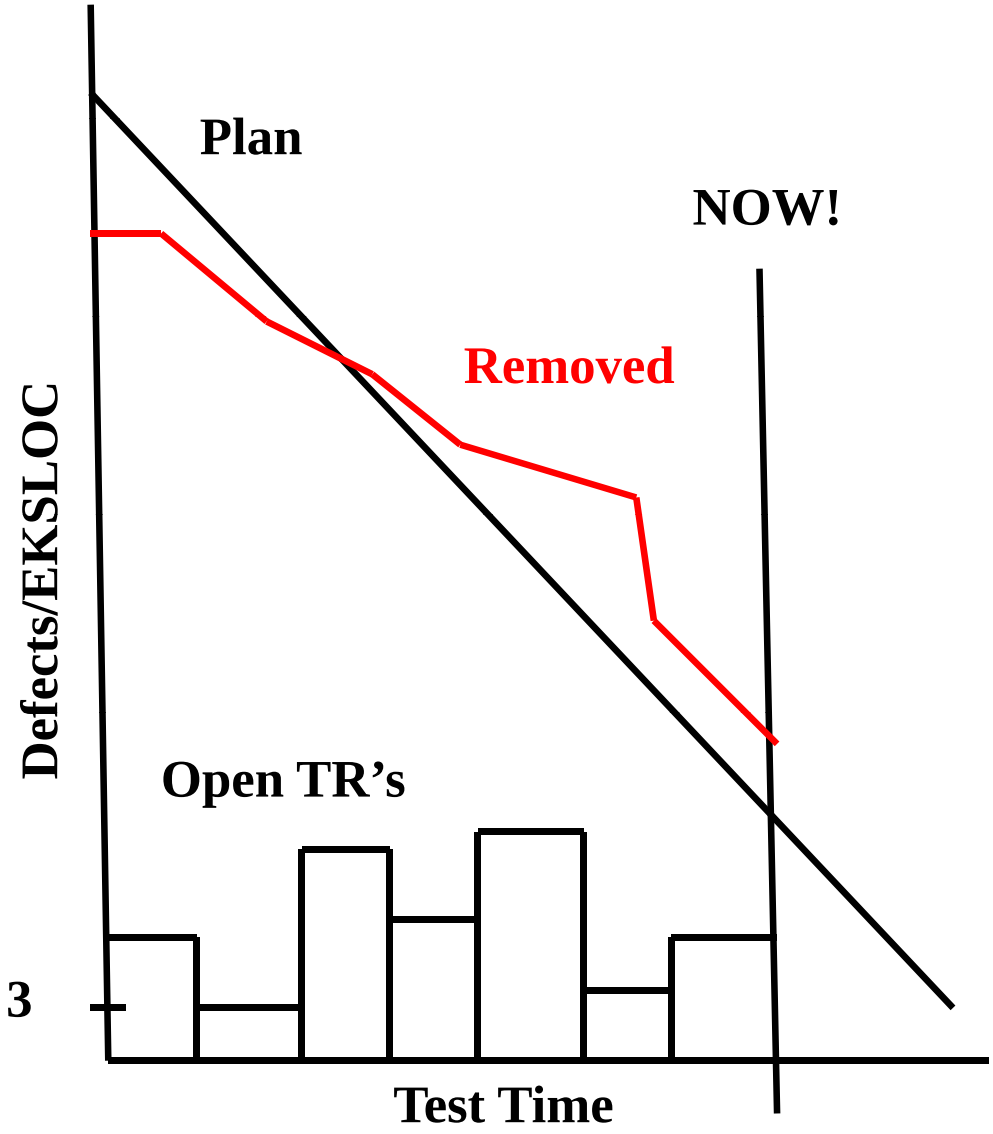
Value-Based Testing



20% of features, provide 80% of value to customer

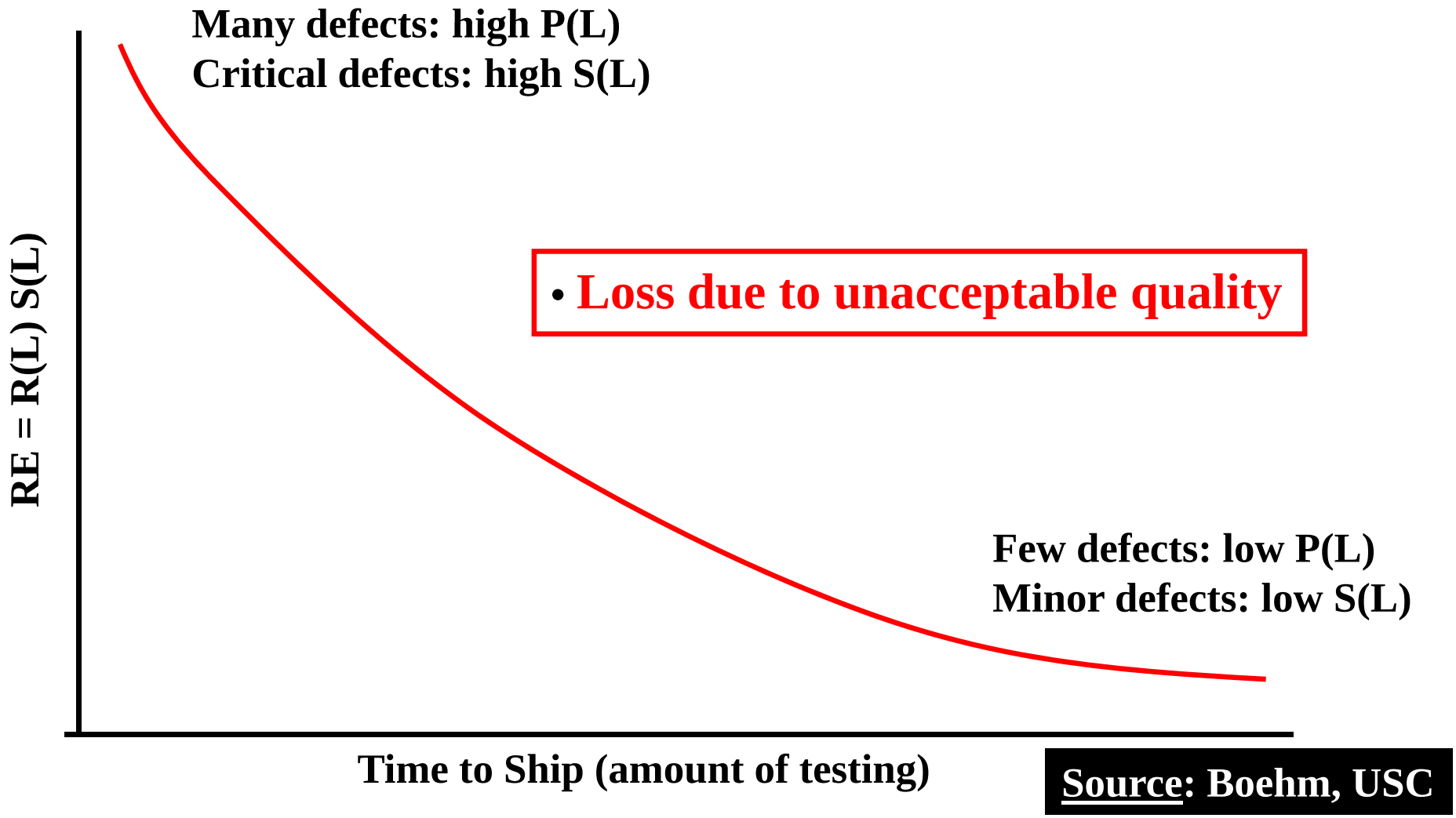
20% of customers are responsible for 80% of business

Defect Removal Model



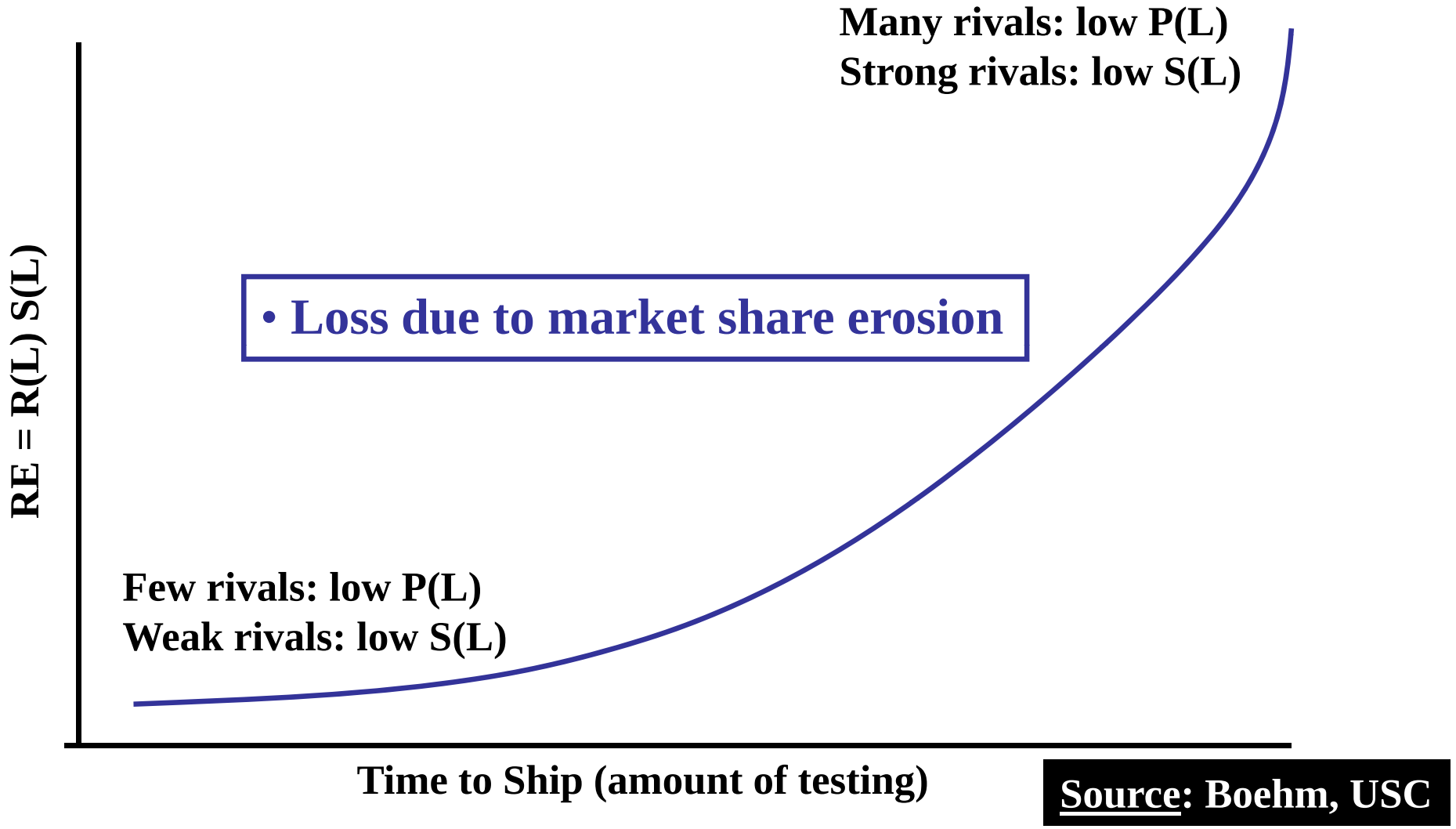
- Set following criteria for release from testing:
  - No critical errors remaining
  - No Trouble Reports open
  - All Trouble Reports (TR's) dispositioned
  - Predicted defects remaining are equal to or less than 3/EKSLOC
- All requirements are satisfied for build, release or delivery
  - Verified via traceability matrices

Example RE Profile: Time to Ship



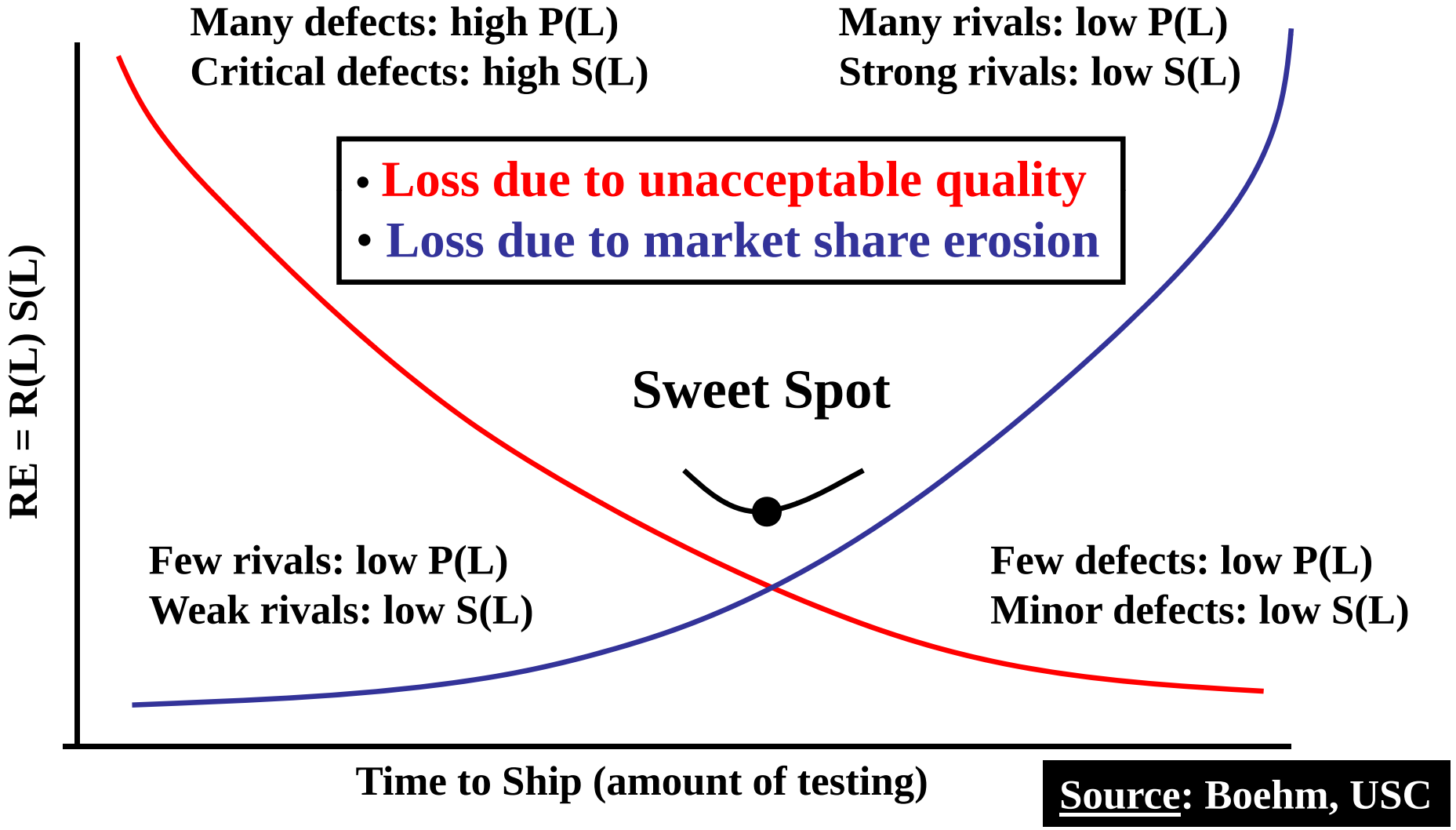
Source: Boehm, USC

Example RE Profile: Time to Ship



Source: Boehm, USC

Example RE Profile: Time to Ship



Source: Boehm, USC

Additional Value-Based Propositions Examples

- About 75% of the work performed in maintenance phase involves testing
  - Develop regression test baseline during development for revalidation after changes
  - Automate tests using value-based approach
- Value proposition
  - Cost increase of 10% during development saves 50% during maintenance

Assess alternatives

- Baseline requirements as soon as possible
  - Get well using ECP's
- Baseline requirements when risk is acceptable
  - Implement as many features as budget will sustain

Which approach leads to more business?

Making Needed Improvements

- Requires courage
- Requires “numbers”
- Is not easy – must overcome resistance to change
- Does not require invention
  - What is needed is often obvious
- Requires you to baseline where you are relative to competition
- Forces you to develop a plan
  - That is your takeaway today

Questions or Comments

- Donald J. Reifer

  dreifer@earthlink.net

  Phone: 310-530-4493

When eating an elephant take one bite at a time.

…….Creighton Adams

An elephant is a mouse built to Mil-Spec.

…….Sayings Galore