



The Effectiveness Formula:

*Key to Next Generation
Development*



Latest Acquisition Shift

- Recent acquisition headlines
 - Air Force One -- \$4B Costs too much, should be cancelled
 - Lockheed F35 – Costs too much
 - Trump claims \$600M cut from F35 program
- New acquisition world ahead
 - Development cost and productivity improvement are targets
 - **“Agile”** is latest technology silver bullet



Agile Manifesto

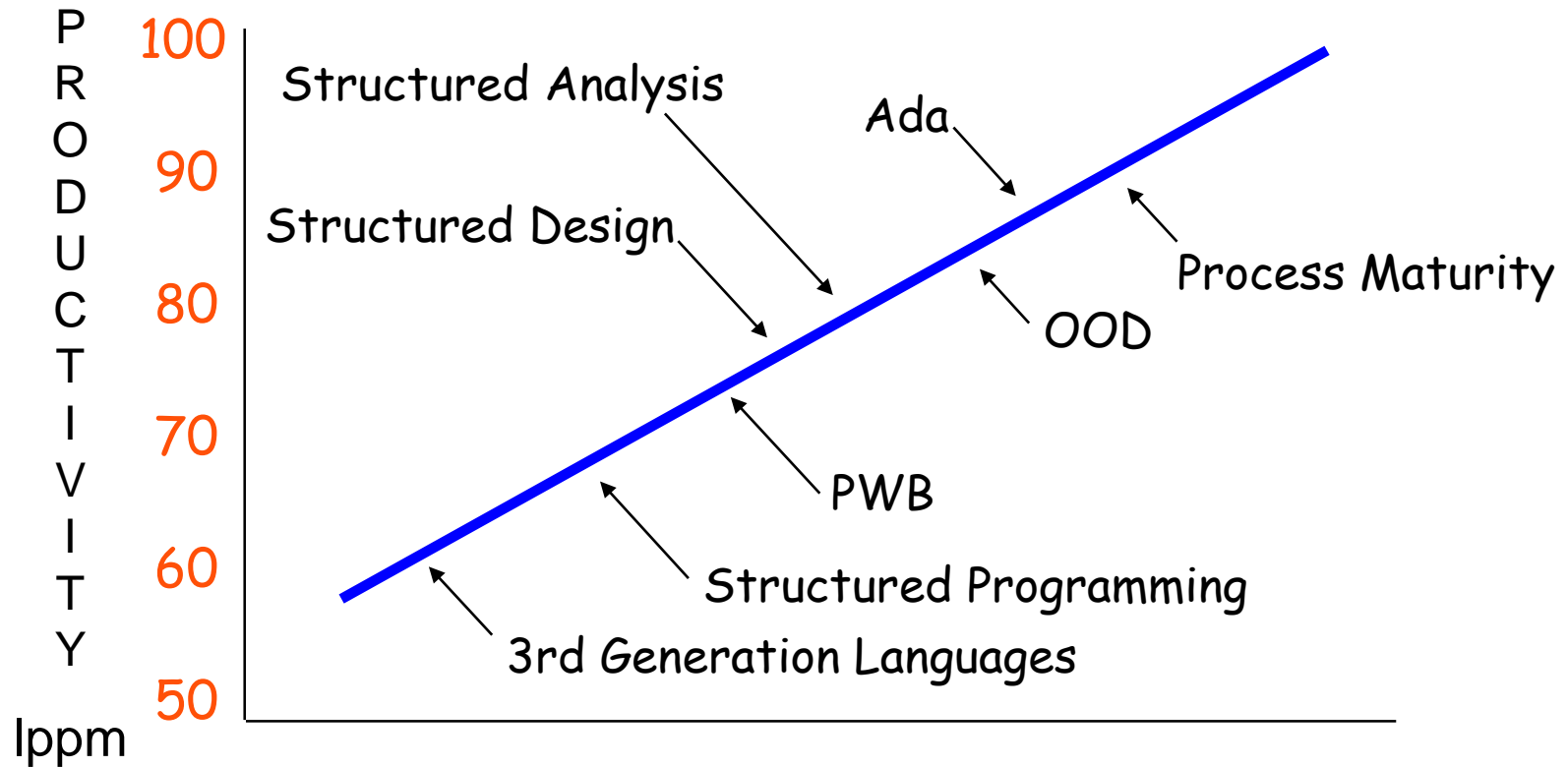
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools.*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation.*
- *Responding to change over following a plan.*

That is, while we value the items on the right, we value the items on the **left** more.



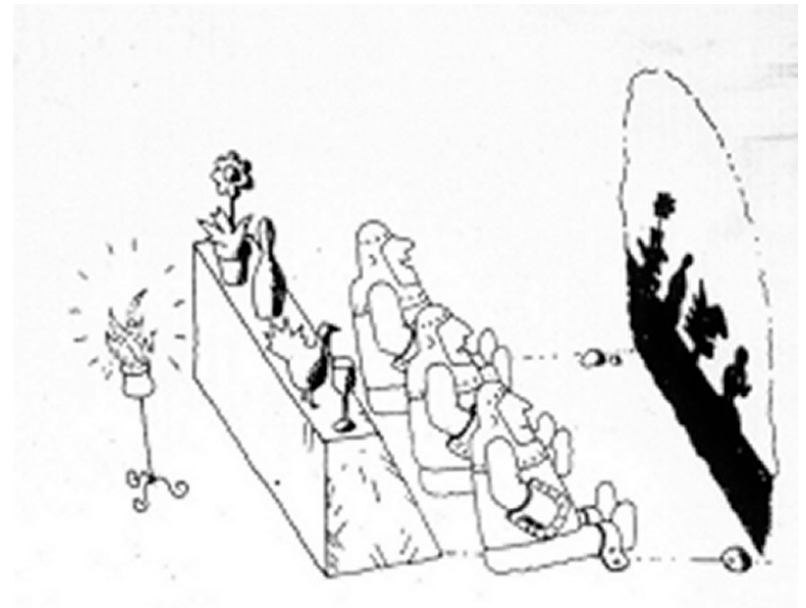
Software Productivity Gains





Plato's Cave

- Perception limited to image on wall
- Traditional mentality
- Reliance on silver bullets
- Resistance to change
- Minimum improvement
- No out-of-box thinking



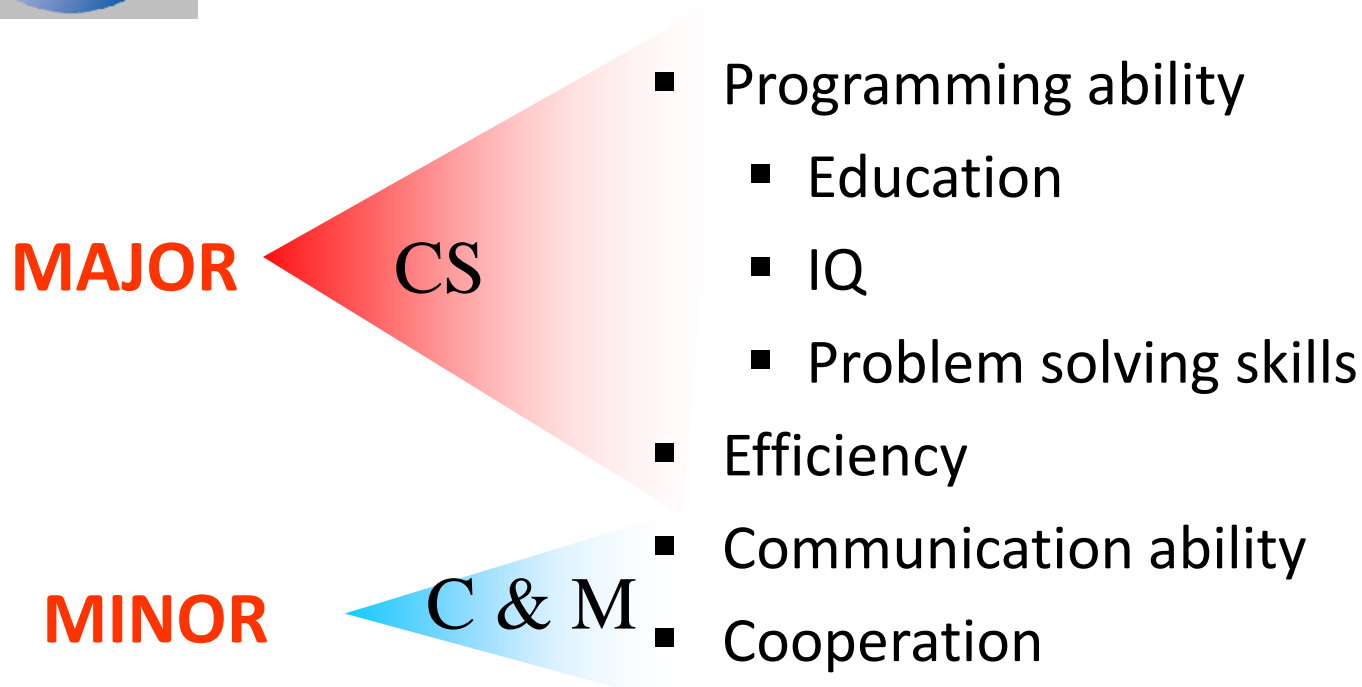


Process Evolution

- Code and Repair
- Waterfall 2167A (1960s)
- Proto typing
- Incremental
- Spiral Development (1970s)
- Rapid Application Development (RAD)
- Capability Maturity Model (CMM) (1990s)
- CMM Integrated (CMMI)
- **Agile**



Traditional Capability Rating



Traditional data does not work for agile development environments



Caverns of Socrates

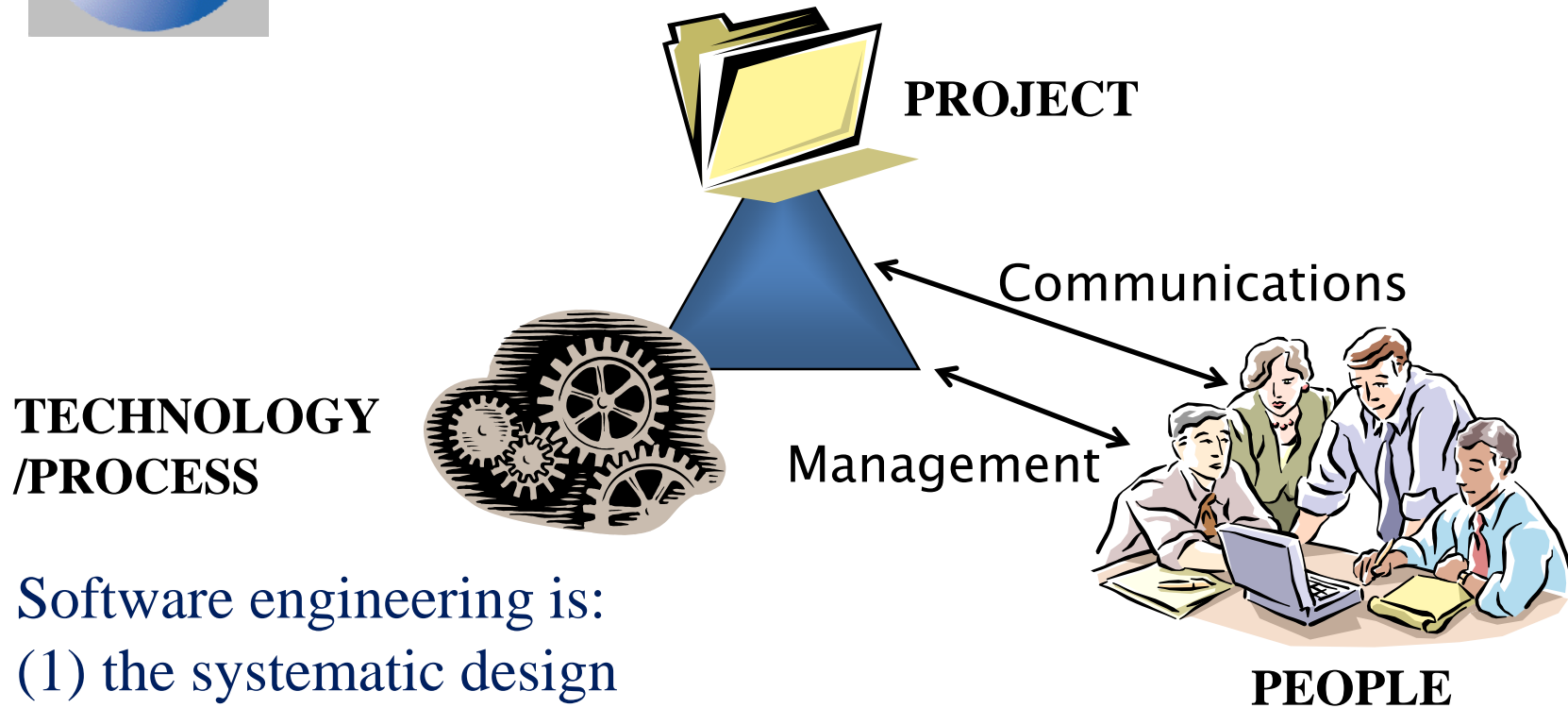
Our job is to escape the cave, look around, then come back and tell others what we have seen . . .

Of course, they won't believe us.

Dennis L. McKiernan, 1995



3 Dimensions of Management



Software engineering is:
(1) the systematic design
and development of
software products (2) the
management of the software
process.



Effectiveness Formula

$$E = C[M(CS)]$$

where

E = Effectiveness

C = Communication skills (0 – 1)

M = Management concept awareness (0 – 1)

CS = Computer science technical ability (0 – 1)

Jensen and Tonies, *Software Engineering*, PH, 1979

Initial Goal: Development environment model

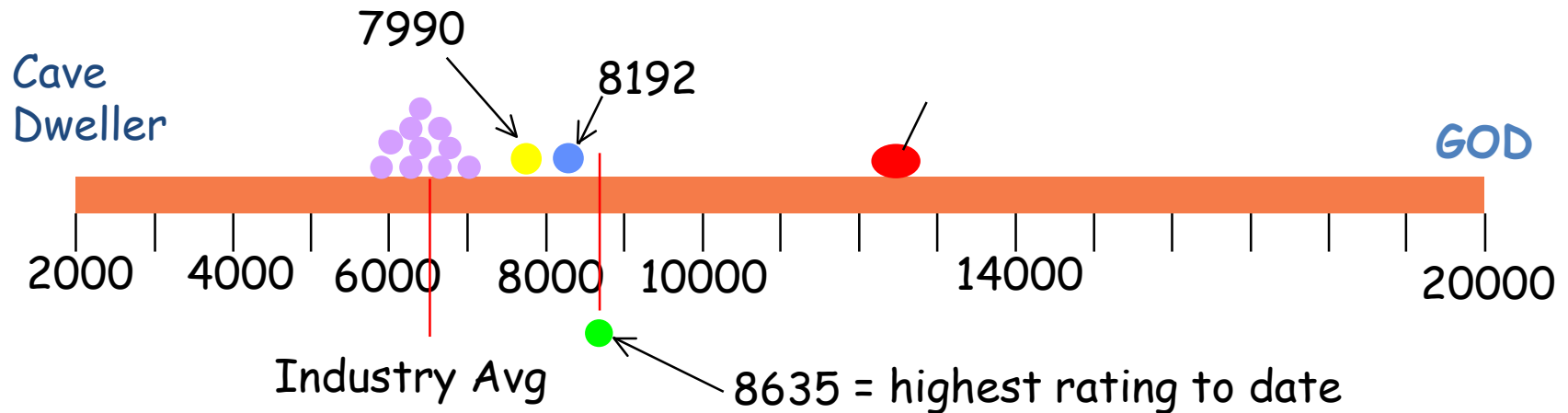


History Contributions

- Low productivity improvement over the last 3 decades has been primarily driven by technology
- Productivity/Effectiveness is a function of 3 attributes: communications, management, and technology
- Communications and management are **KEY** productivity drivers
- Effectiveness Formula applies to almost all development environments



The Ctb Calibration Surprise



- Same application, same organization, same facilities, same people
- Two projects far outside grouping
- Traditional definitions could not explain outriders
- Management effectiveness was the differentiator



Hawthorne Effect

- 1924 Productivity Experiment
 - *Western Electric Company, Hawthorne, Illinois*
 - *Conducted by Harvard Business School*
- Initially designed to study illumination productivity effects
- Illustrated importance of worker's attitudes and feelings on performance
- People driven by esteem and self-actualization, not by physiological and safety needs



Lockheed Skunk Works™

- Unofficial name given to Lockheed Advanced Development Projects Unit managed by Kelly Johnson – dates back to the period around WWII.
- Dispenses with both physical and nonphysical walls
- Known for producing the P-80 fighter aircraft prototype (XP-80) in only 137 days.
- Johnson: “*We are defined, not by the technologies we create, but by the process in which we create them*”



Generic *Skunk Works* Defined

- A small group of experts who move outside an organization's mainstream operations in order to develop a new technology or application as quickly as possible without the organization's bureaucracy or strict process application.
- Creative free thinkers without conventional boundaries
- Physically open workspace environment that encourages intra-team access and communications
- Tools and processes tailored to the project's requirements.
- Minimize development risk while maintaining the greatest possible **agility and creativity**.



Cube Farm

- Counterexample to Skunk Works environment
- Violates all rules for a productive environment in terms of both communications and collaboration
- Raises all barriers that block effective communication
- Devolved from the Action Office originally used by the Lockheed Skunk Works
- Most widely used software development environment today.



Management Research Focus

- General management concepts, 1911
Management philosophy: Plan, Organize, Command, Coordinate, Control
- Hawthorne experiment (1924-1932)
- People impacts, Mayo, 1933
- Lockheed Skunk Works, 1940s
- Theory X/ Theory Y, McGregor, 1960
- CMM, Humphrey, 1989

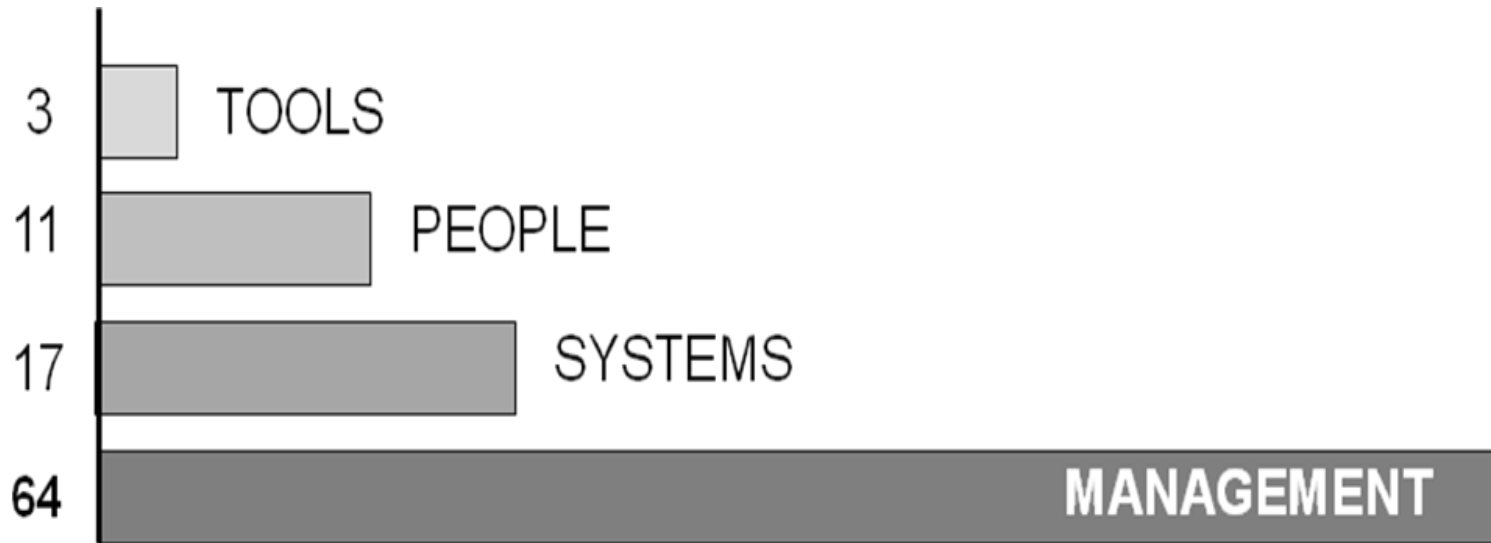


Theory X /Theory Y Concept

Theory X	Theory Y
<ol style="list-style-type: none">1. Work is inherently distasteful to most people.2. Most people are not ambitious, have little desire for responsibility, and prefer to be directed.3. Most people have little capacity for creativity in solving organizational problems.4. Motivation occurs only at the physiological and safety levels.5. Management philosophy: Plan, Organize, Command, Coordinate, Control	<ol style="list-style-type: none">1. Work is as natural as play, if conditions are favorable.2. Self-control is often indispensable in achieving organizational goals.3. The capacity for creativity in solving organizational problems is widely distributed in the population.4. Motivation occurs at the social, esteem, and self-actualization levels, as well as physiological and security levels.



Relative Project Impact



Source: G. Weinberg, Quality Software Management, Vol. 3



1975 Pair Programming Study

- Real time system executive
- 30,000 Fortran source lines
- 6 software components
- 5 two-person teams plus manager
- Average productivity prior to study = 77 LPPM
- Pilot study productivity = **175 LPPM**
- Error rate **<0.1%** of norm



Effective Team Definition

- Team members:
 1. Operate with high degree of interdependence,
 2. Share authority/responsibility for self-management,
 3. Are accountable for collective performance,
 4. Work toward common goal and shared rewards.
- A team has a strong sense of mutual commitment that creates synergy, thus generating performance greater than the sum of performance of its individual members.

Source: businessdictionary.com



Cross Functional Team Study

- Real Time System
- 57,500 Ada Source Lines
- 1 Software Component
- Cross Functional Team Organization
- Average Productivity Prior to Study = ??
- Pilot Study Productivity = 218 LPPM
- Basic Technology Constant = **8635**
(highest recorded in aerospace industry)



Effective Team Environment

- Physically open environment that encourages intra-team access and communication
- Tools and processes adapted to project requirements
- Maintain free flow of information
- Minimize outside interference
- Maximize collaboration and communication



Culture Issues

- Organization Culture
- Comfort Zone
- Confirmation Bias
- K-12 Phenomenon



Organization Culture

- Development organizations have a lot in common
- Technologies
- Physical environment
- Management
 - Hierarchical pyramid
 - Classical 20th century approach
 - *Plan, organize, staff, control, direct*
- Basic Technology Constant ?



Comfort Zone

- Definition: *a place or situation where one feels safe or at ease and without stress.*
- Mental boundary or behavioral state
- Provides an *unfounded* sense of security
- Person tends to stay within the “Zone”
- Stepping outside boundary raises anxiety level and stress response, a danger zone



Confirmation Bias Examples

- Structured programming (1970s)
- GOTO-less programming
- Ada
- Object-oriented development
- Cubicles
- CMMI



CMMI (Process) Impact

*While there are many unique characteristics to software, **they all require more management discipline, not less.** Managers should thus demand detailed plans, tracking systems, and periodic technical and management reviews. **Software management should be entirely traditional, only more so.** Unfortunately, many managers who insist on these items for hardware let their software teams get by without them.*

Watts S. Humphrey, *Managing the Software Process*, (Addison-Wesley, Boston, MA), 1989



K-12 Contributions

- K-12 ingrains inefficiency and waste
- Discourages learning and personal improvement
- Suppresses communication skills
- Workers are inherently lazy and must be closely supervised
- Impedes team formation and behavior
- Ideal conditioning for cubicle behavior

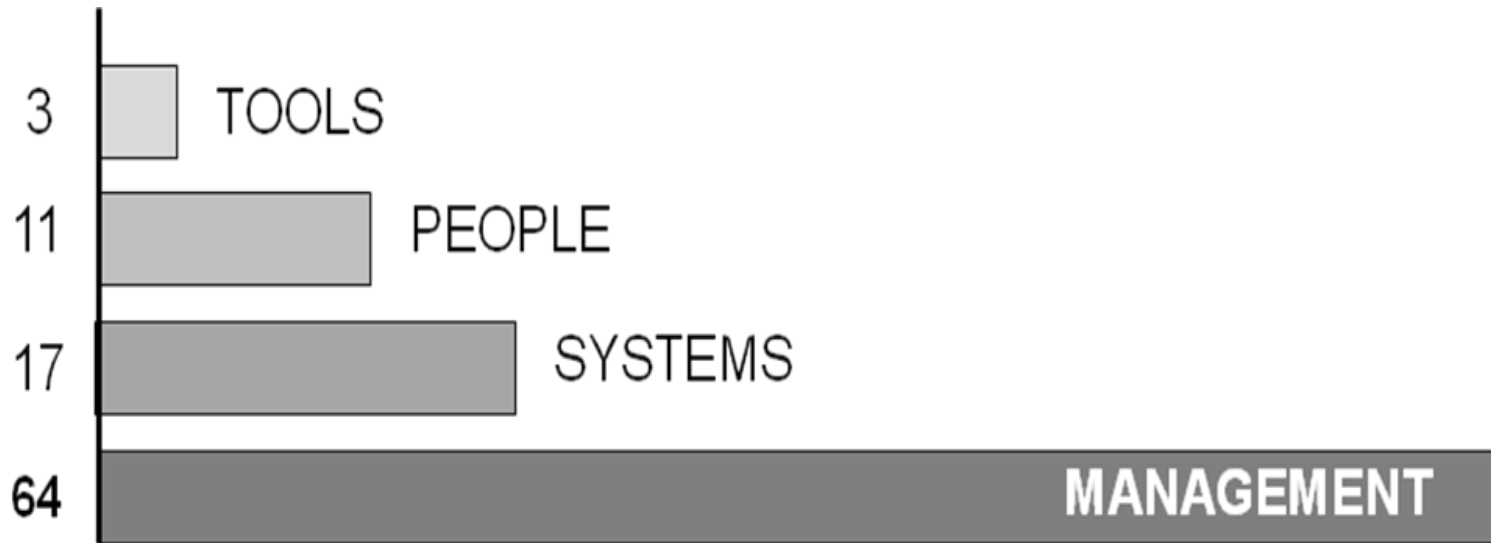


Status Quo Pressure

- Cultural resistance to change is still strong
 - Comfort zone
 - Confirmation bias
 - K-12 phenomenon persists
- Reliance on “Silver bullets”
- Management school support of only *plan, organize, command, coordinate, and control* functions
- CMMI (1985)



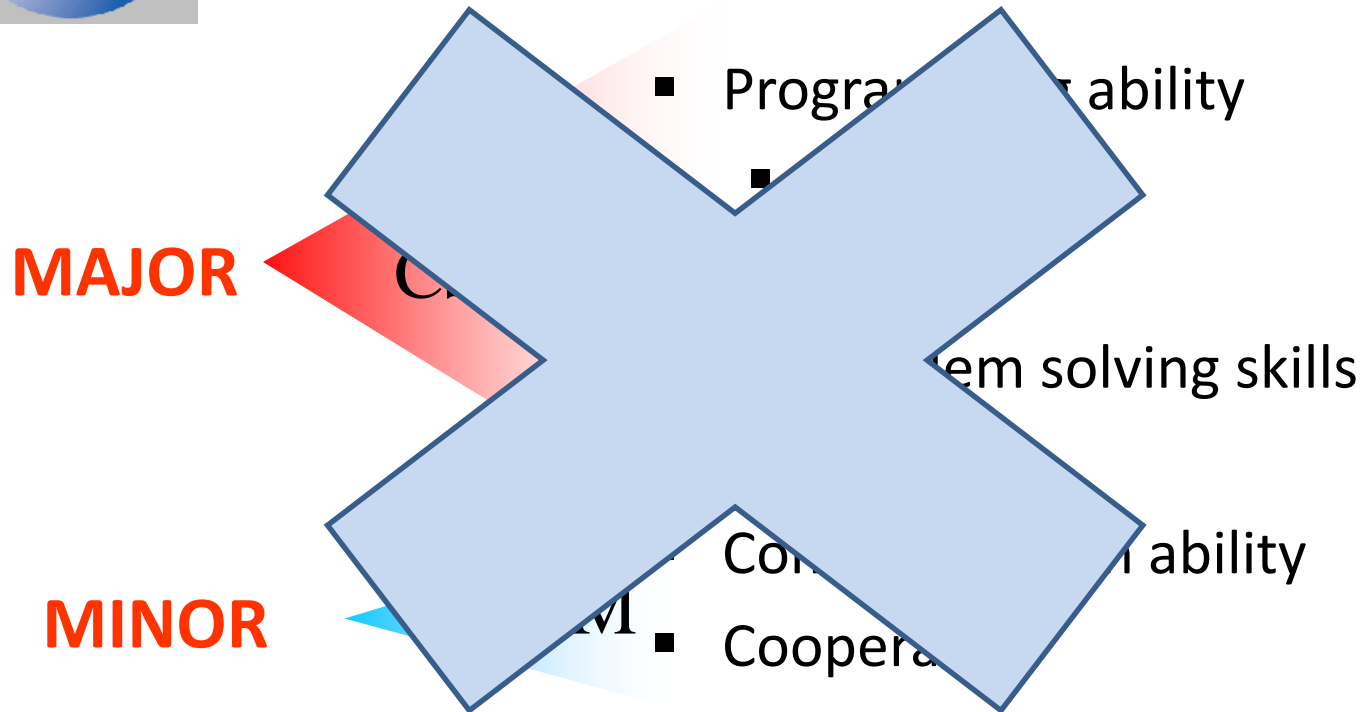
Relative Project Impact



Source: G. Weinberg, Quality Software Management, Vol. 3



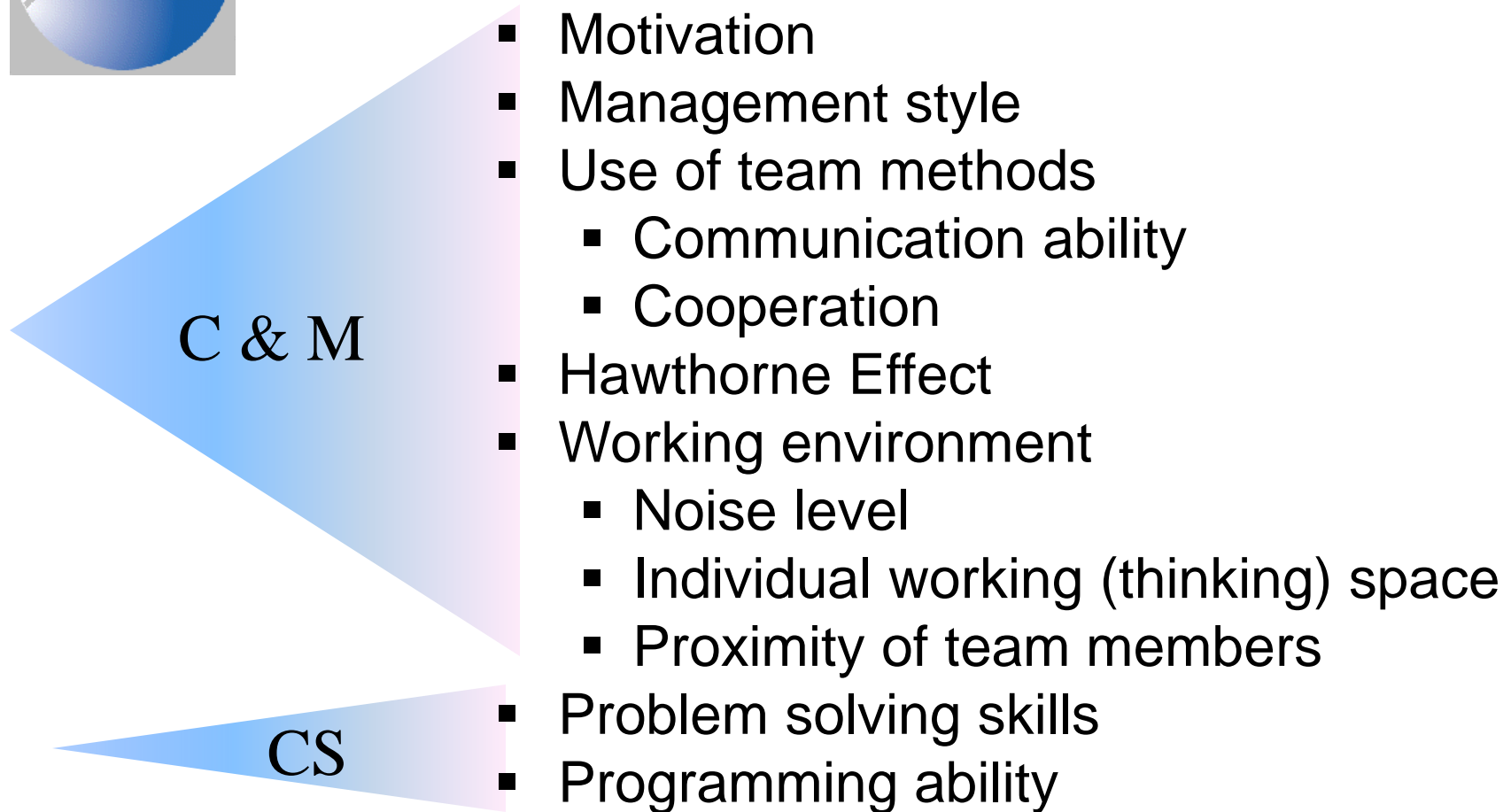
Traditional Capability Rating



Traditional data does not work for agile development environments



Effective Capability Rating





Effectiveness Formula Redux

$$E = CM (CS)$$

where

E = Effectiveness

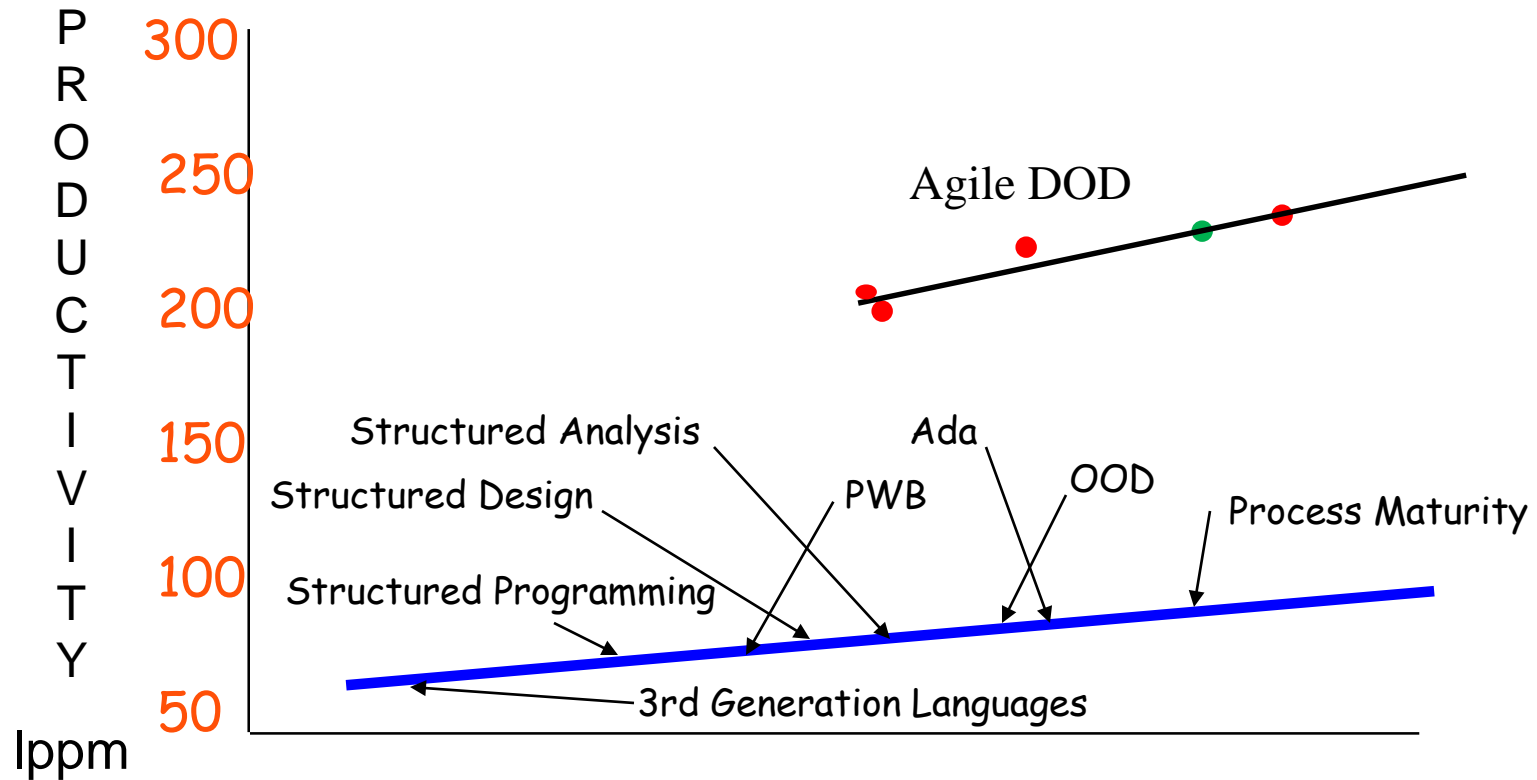
CM = Communications/Management ability (0 – 1)

CS = Software Engineering technical ability (0 – 1)

- Forty five years of validation data !!
- Data does not permit communication – management split



Software Productivity Gains





Development Component:		Task Name		Basic Estimate	
Complexity, D	12	Cte Calculation			
		Experience			
Effective Size		DEXP	1.00	Development Effort	586.5PM
SLOC, New	80,000	LEXP	1.00	Productivity	140.9SLOC/PM
SLOC, Modified	2,000	PEXP	1.00	Requirements Level	8.0Percent
SLOC, Reused	33,000	TEXP	1.00	Requirements Effort	46.9PM
Size, Total	115,000	Support		Integration Level	22.0Percent
Size, Effective	82,650	PVQL	1.00	Integration Effort	129.0PM
		RLOC	1.00	Total Effort	762.5PM
		Environment			
ACAP	1	DVOL	1.07	Development Schedule	26.2MO
AEXP	1.29	RDED	1.00	Requirements Schedule	7.4MO
MODP	1.1	Management			
PCAP	1	MCLS	1.00	Peak development staff	34.5Pers
RESP	1	MORG	1.00	Peak programming staff	18.5Pers
TOOL	1	MULT	1.00	Max staffing rate	26.2Pers/Yr
TURN	0.87	Product			
		DISP	1.05	Maintenance	
Capability Score	0.19	HOST	1.00	Upgrades (ACT)	10 %/Yr
Percentile Rating	29%	MEMC	1.00	Upgrade Effort	87.2PM/Yr
		QUAL	1.03	Knowledge Retention	88.2PM/Yr
Ctb	4,953	RTIM	1.00	Support level	88.2Pers/Yr
		RVOL	1.07		
Templates		SECR	1.00		
Management		SPEC	1.12		
	3	TEST	1.05		
Product		TIMC	1.00		
	13	TVOL	1.00		
		Cte	3,402		



Effectiveness Value Rating



Eff Value	Percentile of Industry	Ctb Value
0.1	5.0	
0.17	22.4	4570
0.2	35.2	
0.23	50.0	5707
0.28	57.9	6735
0.3	61.0	7000
0.31	62.5	7350
0.4	75.2	8630
0.45	81.0	9587
0.5	86.0	
1.0	99.9	



The Next Generation?

- The next generation of software development is already here
 - First organization experiments in “modern management in 1910s (Hawthorne Effect)
 - First evidence of an organization shift is traceable to the mid 1940s (Lockheed Skunk Works®)
- Symptoms of next generation shift
 - Agile Manifesto (2001)
 - Herman Miller Co. introduces new office evolution including movable walls and open workplace forms (2013)
- **New stress on acquisition cost and productivity**



Caverns of Socrates

Our job is to escape the cave, look around, then come back and tell others what we have seen . . .

Of course, they won't believe us.

Dennis L. McKiernan, 1995



Productivity?

