Defect Collection & Analysis –
The Basis of Software Quality Improvement

Joe Schofield,

Sandia National Laboratories

Albuquerque, N. M.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Defect Collection & Analysis –
The Basis of Software Quality Improvement

Joe Schofield,

Sandia National Laboratories

Albuquerque, N. M.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

A Quick Look Back & Update on Recent IFPUG / ISMA Presentations

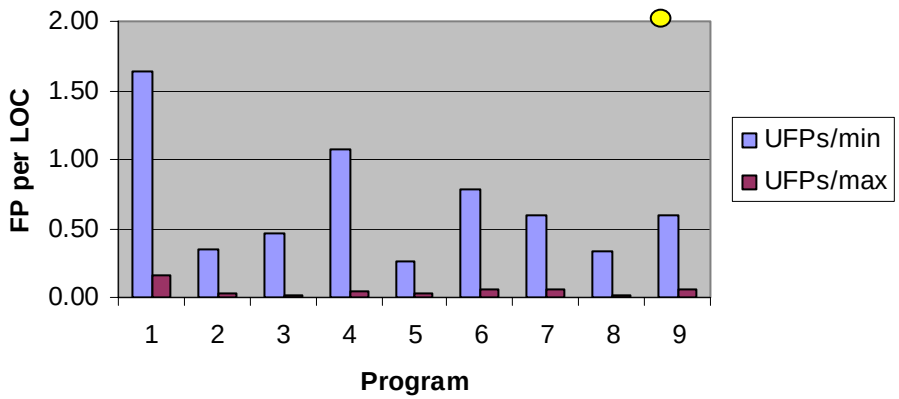
2002	**Counting KLOCs – Software Measurement's Ultimate Futility (I can't do this anymore, or who am I fooling?, or why not count ants?)** ***The Statistically Unreliable Nature of Lines of Code;* CrossTalk, April 2005** ***A Practical, Statistical, and Criminal Look at the Use of Lines of Code as a Software Sizing Measure* ; N.M. SPIN; March, 2004**
2003	**Amplified Lessons from the Ant Hill –** **What Ants and Software Engineers Have in Common** ***Lessons from the Ant Hill - What Ants and Software Engineers Have in Common*; Information Systems Management, Winter 2003**
2004	**Applying Lean Six Sigma to Software Engineering** ***When Did Six Sigma Stop Being a Statistical Measure?; CrossTalk, April 2006*** ***Lean Six Sigma - Real Stories from Real Practitioners*; Albuquerque, N.M.; N.M. SPIN; August 2005** ***Six Sigma & Software Engineering: Complement or Collision*; Albuquerque, N.M.; N.M. SPIN; August, 2004**
2005	**Defect Collection & Analysis – The Basis of Software Quality Improvement** ***Defect Management through the Personal Software Process(SM); CrossTalk, September 2003*** ***The Team Software ProcessSM - Experiences from the Front Line*; Software Quality Forum; Arlington, Virginia, March; 2003** ***Measuring Software Process Improvement - How to Avoid the Orange Barrels*; System Development, December 2001** ***Usable Metrics for Software Improvement within the CMM*; Software Quality Forum 2000; Santa Fe, N.M.; April, 2000**

Min and max values for “C” code compared to Function Point size over 9 programs (n = 49)

	*P1	P2	P3	P4	P5	P6	P7	P8	P9	
Min	22	20	15	13	27	23	25	21	25	
Max	221	311	336	289	270	306	242	383	284	
UFPs	36	7	7	14	7	18	15	7	15	
UFPs/min	1.64	0.35	0.47	1.08	0.26	0.78	0.60	0.33	0.60	
UFPs/max	0.16	0.02	0.02	0.05	0.03	0.06	0.06	0.02	0.05	
Variance Range	10.05	15.55	22.40	22.23	10.00	13.30	9.68	18.24	11.36	14.76

2002

Largest min to max variance is > 22, smallest is almost 10, average is almost 15.



Note that in these three examples, variance and averages increased as the population increased.

Software engineers are smarter than ants, right?

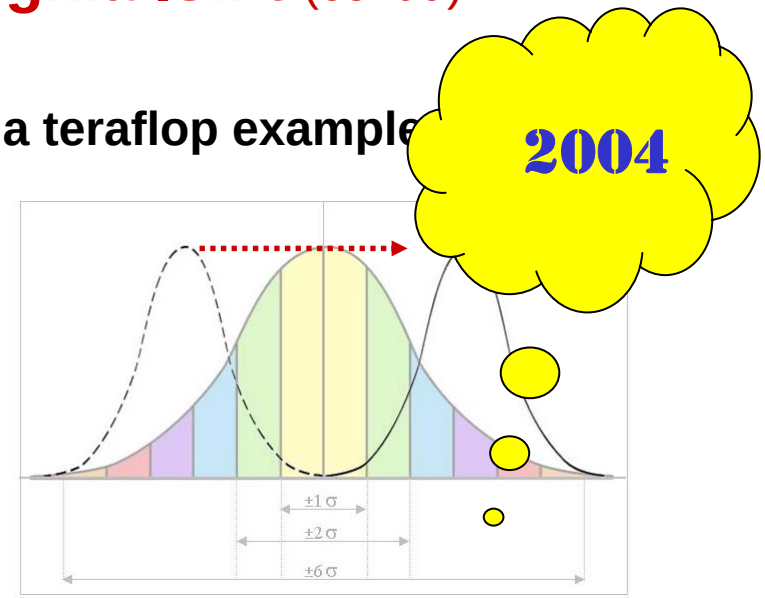
Observation: When ants underestimate the size of a job, they compensate with waves of more ants. Most software projects cannot afford this tactic.

Lesson: Use reliable sizing measures like Function Points to assess progress. Avoid the practice of counting lines of code as a measure of size or progress.

Reference: *A Practical, Statistical, and Criminal Look at the Use of Lines of Code as a Software Sizing Measure*, Schofield, Structured Development Forum, March, 2003

When Lean Six Sigma Isn't (cont'd)

"What if" the *sigma shift* went to the right – a teraflop example



TeraFlops machine

1T floating point operations instructions per second =
3 defects per 100 seconds =
108 defects per hour =
18,144 per week =
943,488 DEFECTS per year =
50M+ a year at "shifted 6 sigma" (4.5 sigma)
(these numbers are rounded down)

Who can repair / afford / manage that many defects?

[1] **PetaFlops machine**

predicted to be ready by 2005 or 2006
1,000 times faster than a 1TFlop machine =
943,488,000 defects per year @ 7.5 sigma =
50B (that's BILLION) at "shifted 6 sigma"

PETAFLOP Imperative; Informationweek; June 21, 2004; pgs. 55 – 62

[2] **IBM's Gene/L at Lawrence Livermore National Lab operates @ 70.72TF**
IBM will increase the speed to 360 TF in 2005

U.S. Regains Top Supercomputer Spots; Informationweek; November 15, 2004; pg. 28

(Back to Defects . . .)
The Business Case for Defect Removal

Software defects cost the U.S. $59.6B a year[1]

38 percent of polled organizations have no SQA program[2]

Software technicians in Panama are charged with second degree murder after 27 patients received overdoses of gamma rays; 21 have died in 40 months[3]

BMW, DaimlerChrysler, Mitsubishi, and Volvo experience product malfunctions (engine stalls, gauges not illuminated, wiping intervals, wrong transmission gears) due to software[4]

A 2002 GAP report showed that spreadsheet errors at NASA contributed to a $644M misstatement in 1999[5]

SPAM will cost the world $50B in lost productivity and other expenses . . . according to Ferris Research[6]

Medical staff report 38 percent defect rate while using computerized physician order entry (CPOE) systems in determining low dose for infrequently used medications.[7]

The FBI's $170M virtual case file project went through 10 program managers before being cancelled.[8]

[1] Informationweek, *Behind the Numbers*, March 29, 2004; pg 94
[2] CIO, *By the Numbers*, December 1, 2003, pg 28
[3] Baseline – The Project Management Center, *We Did Nothing Wrong*, March 4, 2004
[4] Informationweek, *Software Quality*, March 15, 2004; pg 56
[5] CIO, *Essential Technology*, May 15, 2005; pg 74
[6] Informationweek, February 28, 2005; pg 18
[7] CIO, *Medication Systems*, June1, 2005; pg 28
[8] CIO, *Why the G-Men Aren't IT Men*, June15, 2005; pg 44

Basic Measures for Defects

***Defect* – a product anomaly such as an omission or imperfection; faults in software sufficiently mature for test or operation. Adapted from the IEEE 982.1**

Defect Attributes:

Defect injection

Defect detection

Defect removal cost

Defect type

Defect status

Defect severity

Peer Review times

Defect repair cost

Derivable Defect Measures:

Defect per Function Point

Average cost of defect by phase

Average cost of defect by work product

Defect leakage

Predicted number of defects (historic DFP)

Cost of Peer Reviews

Cost of defect removal

Cost of testing

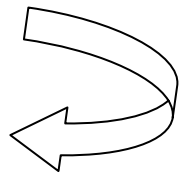
ROI for defect removal (Cost of prevention and removal vs. cost of repair)

Applying DMAIC (Six Sigma) to Defect Data

Actual cost benefit figures for software development

Measures:

Define (opportunities)	**An Organization Definition**
Measure (performance)	**Peer Reviews & Defects**
Analyze (opportunity)	**Defect List & Analysis**
Improve (performance)	**Process Focus & Change**
Control (performance)	**Sustained Measurement & Improvement**



Whats ***Hows***

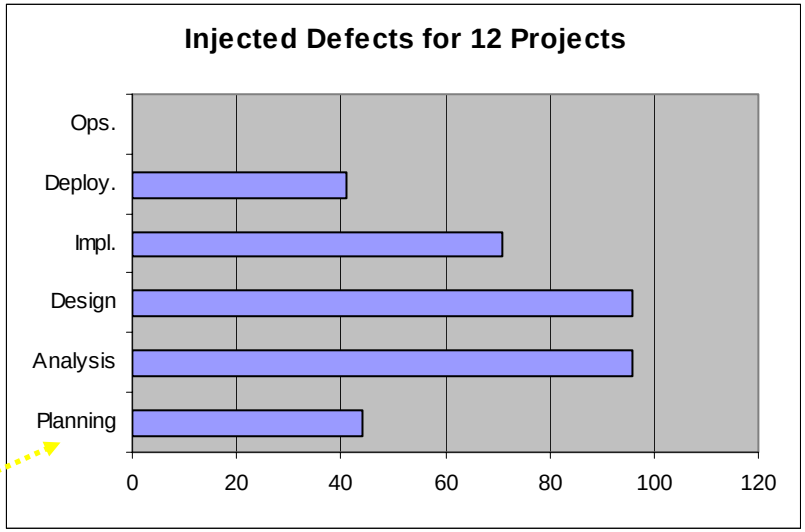
Applying DMAIC to Defect Data (cont'd)

Required items are **bold.**

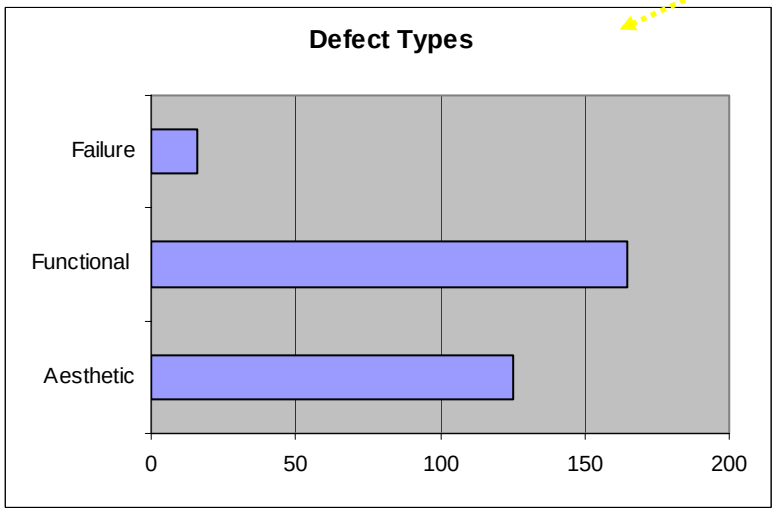
Attribute	Value	
Discovered By	Change Request	Peer Review Item
Detection Phase	Planning	
Injection Phase	Planning	
Defect Type	Completeness	
Defect Severity	Aesthetic	
Cost to Repair		
Description/Class		
Disposition		

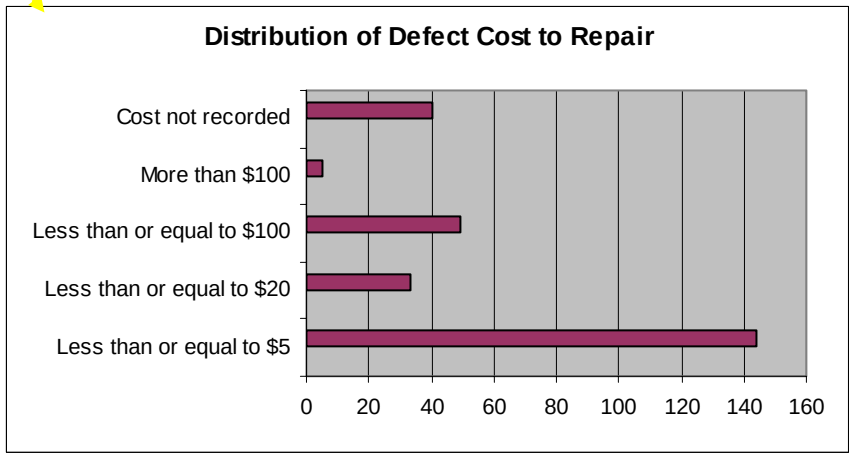
Submit Reset

Measure / Record



Analyze





Applying DMAIC to Defect Data (cont'd)

Modify or Input Defects

SILC Review Phase	Discovered By	Defect Type	Total Defects	Total Cost	Cost Per Defect
Planning	Change Request	Completeness	20	707	35.35
Planning	Peer Review	Completeness	91	3867	42.49
Planning	Peer Review	Consistency	21	667	31.76
Planning	Peer Review	Corrective	33	1481	44.88
Planning	Test Plan	Completeness	1	4	4.00
Analysis	Change Request	Completeness	6	180	30.00
Analysis	Change Request	Corrective	6	60	10.00
Analysis	Peer Review	Completeness	124	2900	23.39
Analysis	Peer Review	Consistency	103	1968	19.11
Analysis	Peer Review	Corrective	109	1890	17.34
Design	Change Request	Completeness	3	160	53.33
Design	Change Request	Corrective	4	170	42.50
Design	Peer Review	Completeness	265	7406	27.95
Design	Peer Review	Consistency	59	1313	22.25
Design	Peer Review	Corrective	162	2054	12.68
Design	Test Plan	Completeness	2	8	4.00
Design	Test Plan	Consistency	1	6	6.00
Design	Test Plan	Corrective	4	124	31.00
Implementation	Change Request	Completeness	2	80	40.00
Implementation	Change Request	Corrective	8	1337	167.13
Implementation	Peer Review	Completeness	63	2125	33.73
Implementation	Peer Review	Consistency	55	1909	34.71
Implementation	Peer Review	Corrective	76	2572	33.84
Implementation	Test Plan	Completeness	36	3801	105.58
Implementation	Test Plan	Consistency	15	1146	76.40
Implementation	Test Plan	Corrective	85	3151	37.07
Deployment	Change Request	Corrective	4	67	16.75
Deployment	Peer Review	Completeness	29	200	6.90
Deployment	Peer Review	Consistency	1	4	4.00
Deployment	Peer Review	Corrective	7	38	5.43
Operational	Change Request	Completeness	5	408	81.60
Operational	Change Request	Consistency	4	195	48.75
Operational	Change Request	Corrective	12	1215	101.25
Operational	Test Plan	Corrective	11	1319	119.91

Modify or Input Defects

Defect summary by Where discovered and How

Find

Remove

Prevent

Applying DMAIC to Defect Data (cont'd)

Find

Remove

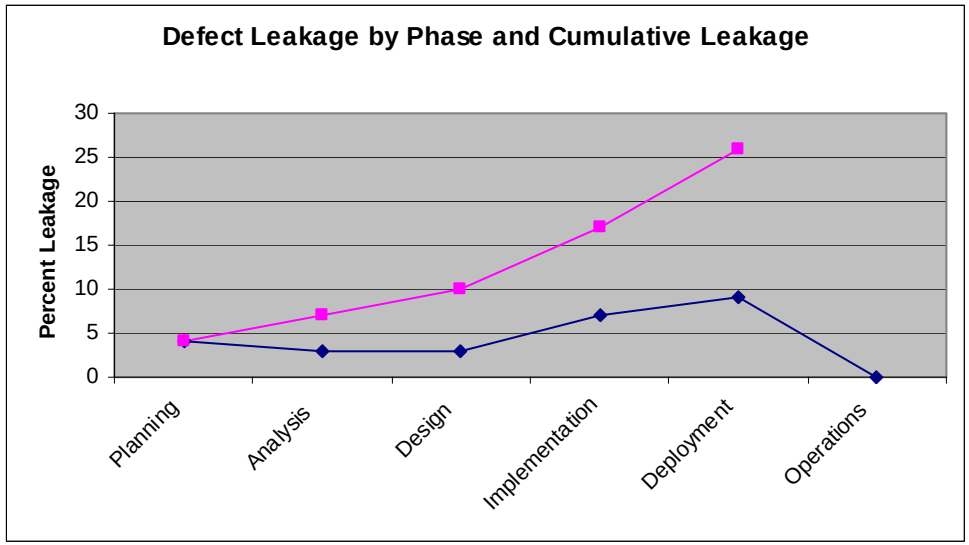
Prevent

Artifact Reviewed	Total Defects	Total Cost	Cost Per Defect
External Interfaces Definition	94	1612	17.15
Information Model	57	525	9.21
Internal Components Definition	67	1507	22.49
Other: Business Rules	5	13	2.60
Other: Business Rules and Use Cases	33	122	3.70
Other: CSA Administrator Documentation	2	20	10.00
Other: CSA User Documentation	2	20	10.00
Other: EP Interim Solution: Source Code	13	308	23.69
Other: External Interfaces Definition - Admin	5	9	1.80
Other: External Interfaces Definition - Reports	9	29	3.22
Other: External Interfaces Definition - Wizard	4	47	11.75
Other: Information Model. Data Dictionary	7	22	3.14
Other: Interim Solution Test Plan	6	159	26.50
Other: Internal Components Definition--Admin	1	1	1.00
Other: Internal Components Definition--Course	6	266	44.33
Other: RS2 User Process Model	5	70	14.00
Other: RS3 SW Requirements & Design Specification	1	50	50.00
Other: RS3 Software Source Code & Executables	5	610	122.00
Other: RS3 User Process Model	8	165	20.63
Other: RS5 User Process Model	4	14	3.50
Other: RS6 User Process Model	7	300	42.86
Other: RS7 Design	7	200	28.57
Other: RS7 Software Source Code & Executables	3	40	13.33
Other: RS7 User Process Model	8	135	16.88
Other: Software Requirements Specification	27	470	17.41
Other: Test Plan, Information Model, External Interfaces	25	134	5.36
Other: Use Case Diagrams & Textual Use Cases	1	2	2.00
Other: Use Case Model	3	7	2.33
Other: User Documentation	17	101	5.94
Project Plan	129	4005	31.05
Software Source Code & Executables	139	4561	32.81
Test Plan	209	3774	18.06
User Process Model	132	5641	42.73

Defect summary by work product

For defect removal, Tom Glib reports some inspection efficiencies as high as 88 percent. Jones, *Software Quality*, pg 215

Applying DMAIC to Defect Data (cont'd)



Given:

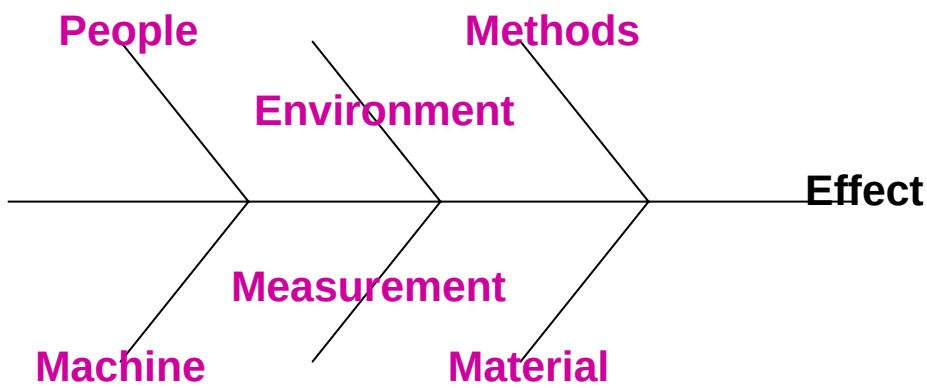
1 Peer Review is performed in Planning

2 Peer Reviews are performed in Analysis

3 Peer Reviews are performed in Design

Look at Planning & Analysis

1) **How are so many defects removed in Implementation?**
2) **Does the organization need more Peer Reviews in Planning & Analysis?**
3) **How effective are Design Peer Reviews?**



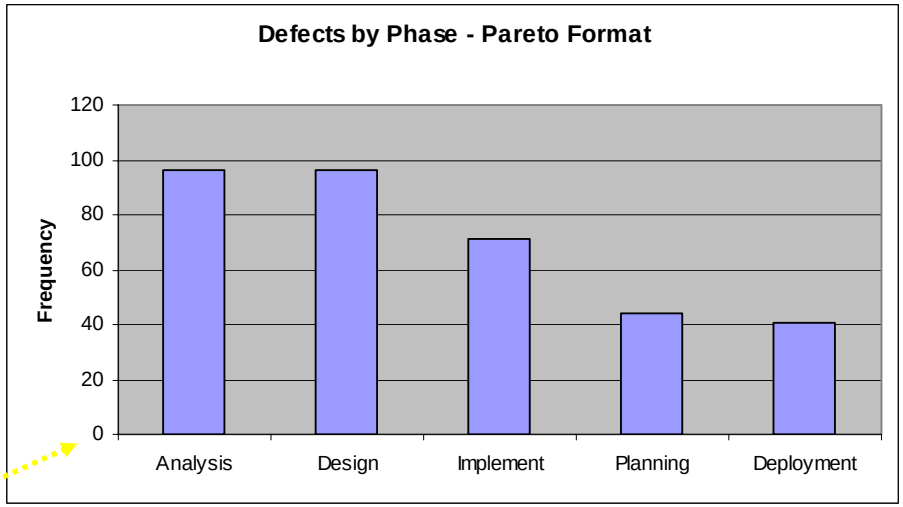
Using Pareto Analysis and Histograms for Defect Data

Required items are **bold.**

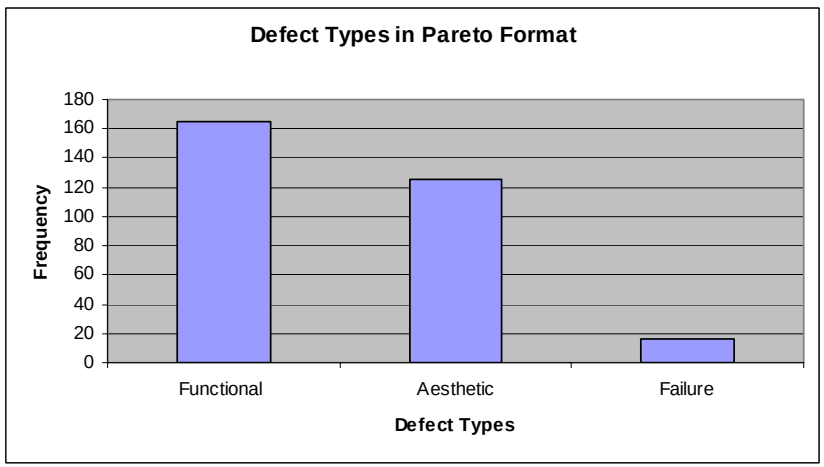
Attribute	Value	
Discovered By	Change Request	Peer Review Item
Detection Phase	Planning	
Injection Phase	Planning	
Defect Type	Completeness	
Defect Severity	Aesthetic	
Cost to Repair		
Description/Class		
Disposition		

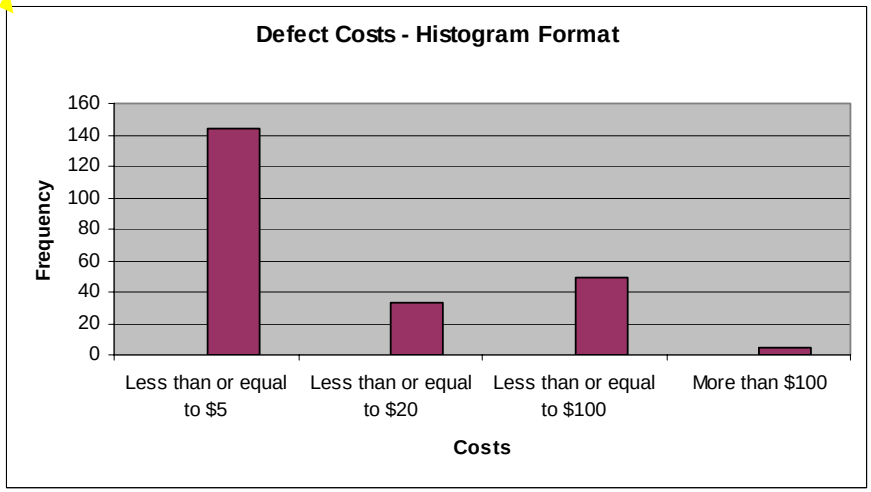
Submit Reset

Captured

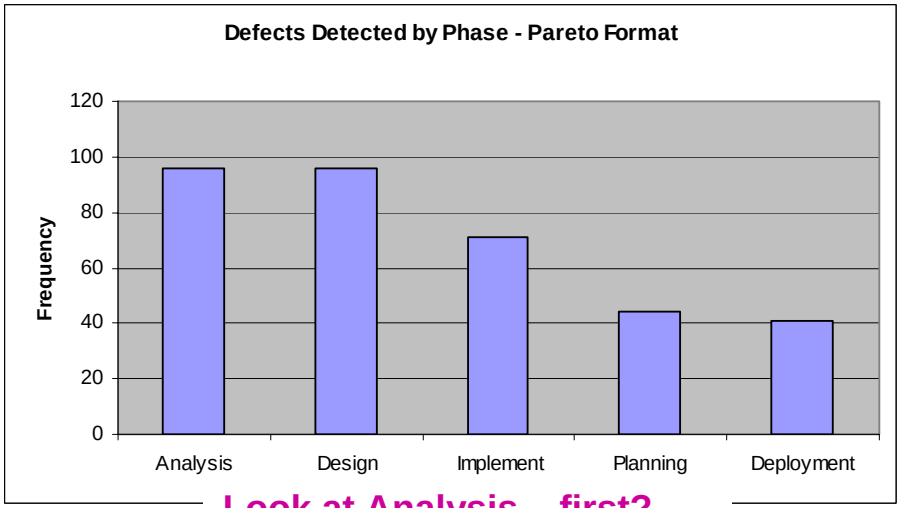


Analyze (Derived)

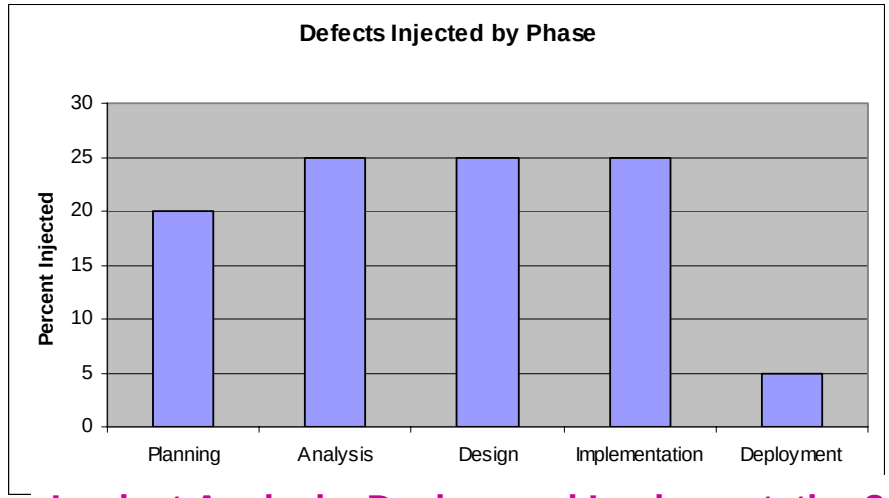




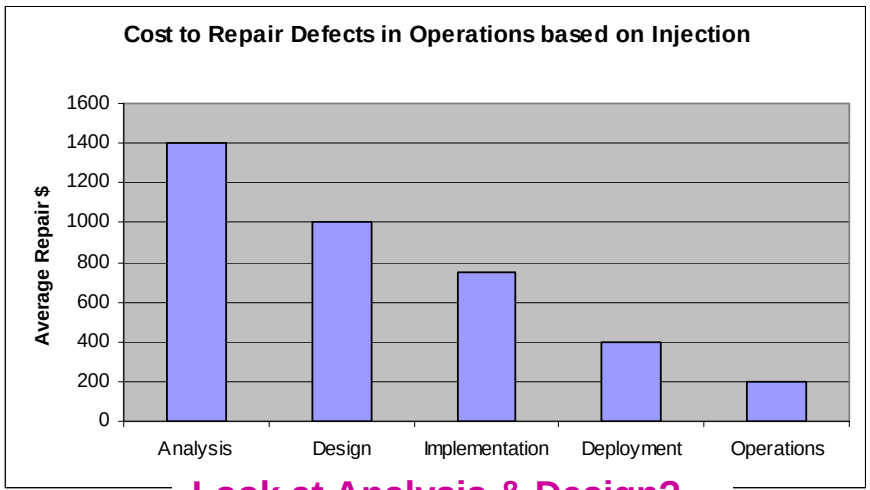
Even Tools Require Thinking



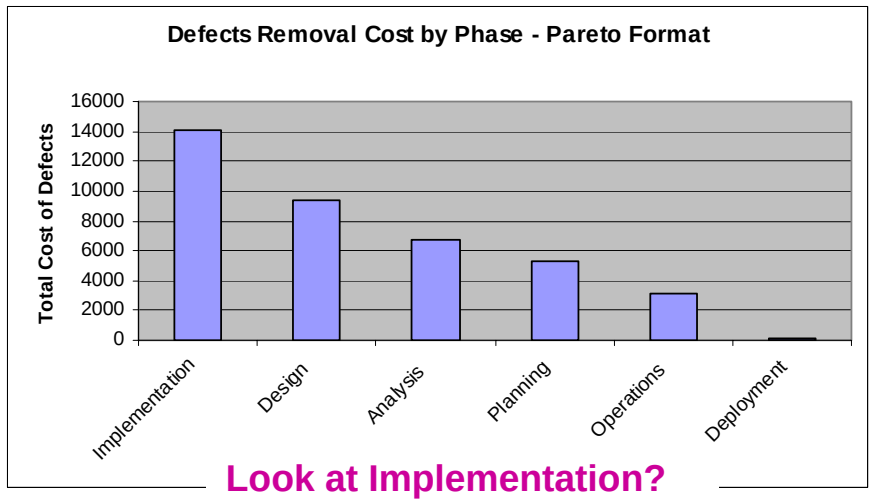
Look at Analysis – first?



Look at Analysis, Design, and Implementation?



Look at Analysis & Design?



Look at Implementation?

A Way to Look at Defects

Phase Injected

Phase Detected	Planning	Analysis	Design	Impl.	Deploy.	Ops.
Planning	109	4	8	8		
Analysis	1	290	2			
Design	3	9	476	2		
Imple.	1	1	13	296		
Deploy.				1	20	
Ops.			3	24	2	30
Total Injected	114	304	502	331	22	30
% leakage	4	3	3	7	9	

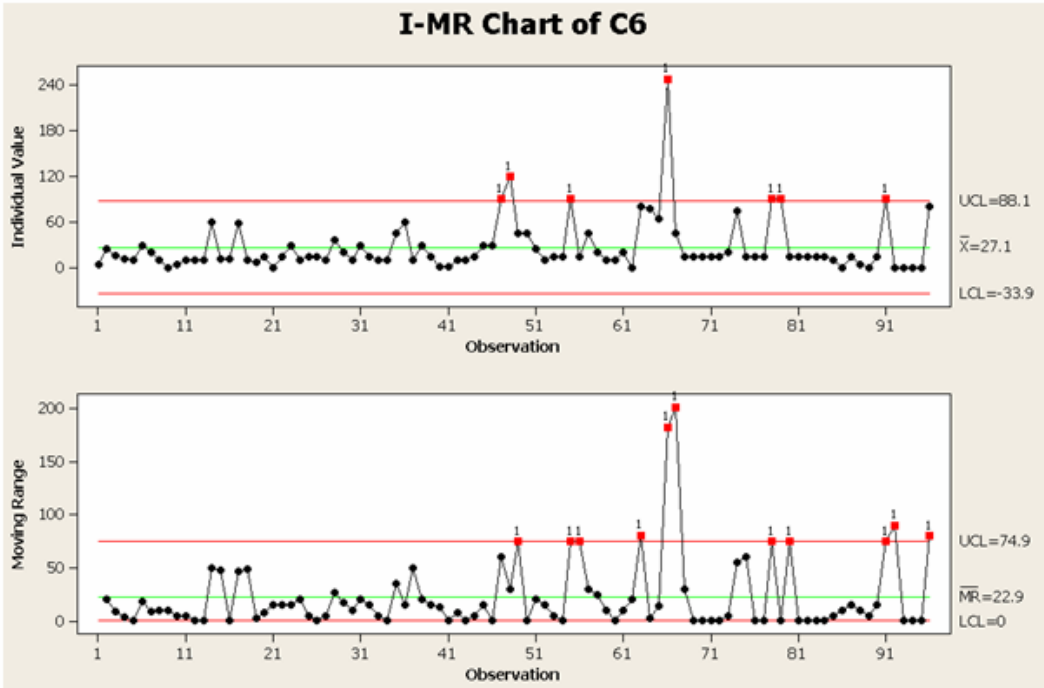
Find

Remove

Prevent

What does this association matrix REVEAL?

Some Statistical Analysis is Required for "Higher Level Maturity"



Find

Remove

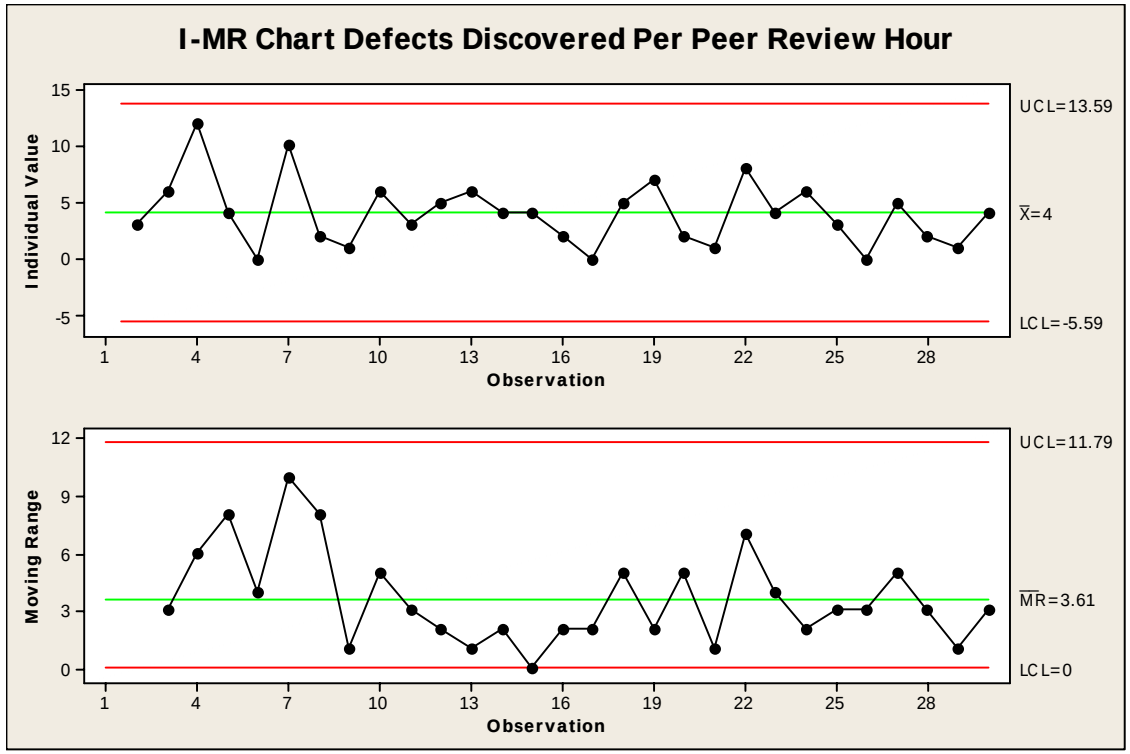
Prevent

Special (Assignable) Cause removal required at CMMI® Level 4

How well the process is performed

Some Statistical Analysis is Required for "Higher Level Maturity"

Balancing the "voice of the customer" with the "voice of the process"



Causal analysis is required at CMMI® Level 5

How well the process is performs

Find

Remove

Prevent

Measurement-related Characteristics of "Higher Level Maturity" Organizations

- **85 percent of high maturity organization align process improvement with TQM initiatives at the enterprise level**
- **85 percent of high maturity organizations have multiple process and quality improvement initiatives**
- **81 percent of high maturity organizations also have ISO 9001 certification**
- **50 percent using CMM, 42 percent using balanced scorecard, 42 percent using LSS, 25 percent using Baldrige**
- **58 percent of high maturity organizations have established formal mentoring programs**
- **15 - 21 percent decrease in effort for a one level increase**

Higher Level Maturity with Statistics course (SEI)

Measurement-related Characteristics of "Higher Level Maturity" Organizations

Higher level maturity organizations:

- **Use centralized measurement**
- **Know why measurement are used, bring tables and charts to interviews and explain them!**
- **Describe "causal" analysis spontaneously during assessment interviews**
- **Have and supply ROI data**
- **Use Pareto analysis**
- **Use control charts**
- **Use Six Sigma**
- **Use *orthogonal defect classification – and this means . . .***
- **Require participation in the SQA Group and / or the SEPG prior to promotion to management**

The 2001 High Maturity Workshop (sponsored by the SEI)

Introducing defect pithy ‘tudes

(a hostile attitude or disposition; that is, not defect friendly)

Defects persist in software; *most* of these come from executing a poor software development process or not executing a good one!

***Defectectomy* – *Surgical* defect removal, often evidenced in peer reviews**

***Defecticide* – the killing off of defects, often evidenced in testing**

***Defect metastasization* – the absence of Defectectomy and Defecticide practices**

Q – Dr = F

Quality without defect (removal) is merely “faking it.”

Closing Reminders

“Higher maturity” organizations MUST use statistical tools for identifying assignable and common cause deviation.

Collecting and statistically analyzing defects is superior to merely counting defects and their origins.

SPC, Pareto, and Histogram charts are simple statistical tools for defect analysis and quantitative improvement.

These same tools can be used in other software development activities (estimated vs. actual, requirements volatility)

The following approaches have been found to be the best in class for defect prevention: JADs during requirements, prototypes during design, and reuse during coding and documentation. **Jones, *Software Quality*, pg 160**

And we do this because . . .

Defects are an aspect of measurement (and part of the CSMS!)

The customer has already paid for a quality product.

Predicting performance should be based on process capability. (voice of the process)

Most organizations are interested in lower operational costs.

Most defects are preventable.

You won't need to clean-up later and you won't need to be as charming!