

V&V Lifecycle Methodologies



By David F. Rico

Overview



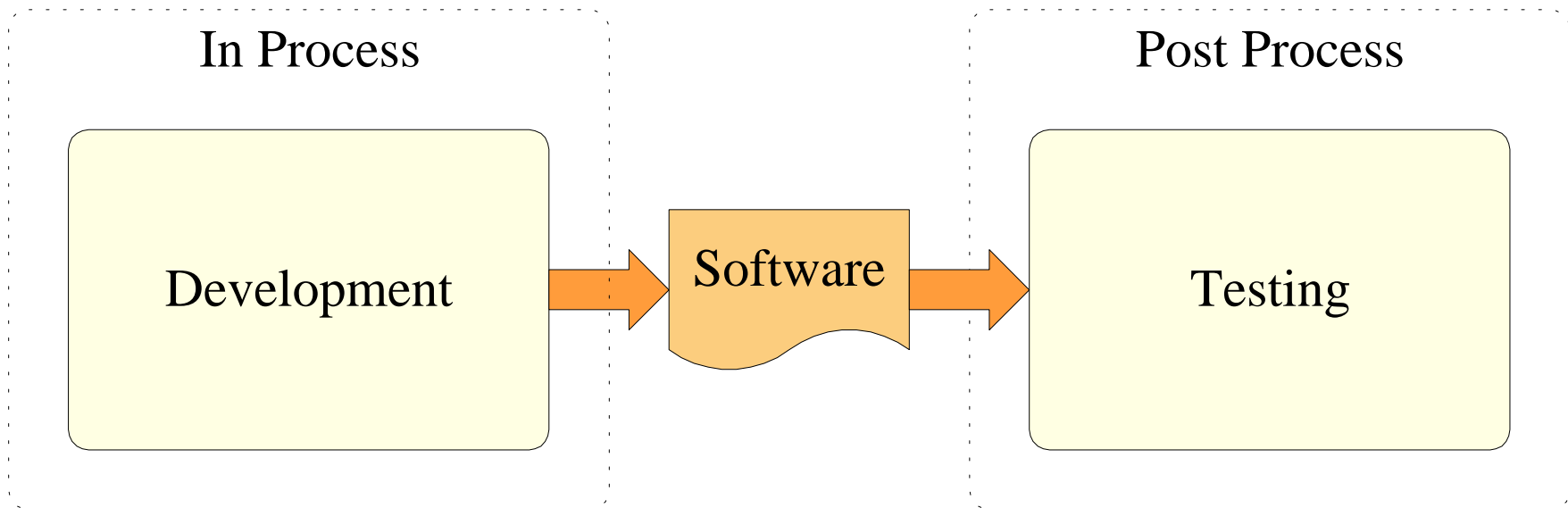
- What is V&V ?
- V&V Approaches
 - Testing (Post Process)
 - Lifecycle Frameworks (In Process)
 - Lifecycle Methodologies (In Process)
- Costs & Benefits
- Myths & Misconceptions
- Conclusion
- Bibliography

What is V&V?



- V&V is the process of determining whether:
 - Requirements for a system or component are complete and correct
 - Products of each development phase fulfill the requirements or conditions imposed by the previous phase
 - Final systems or components comply with specified requirements

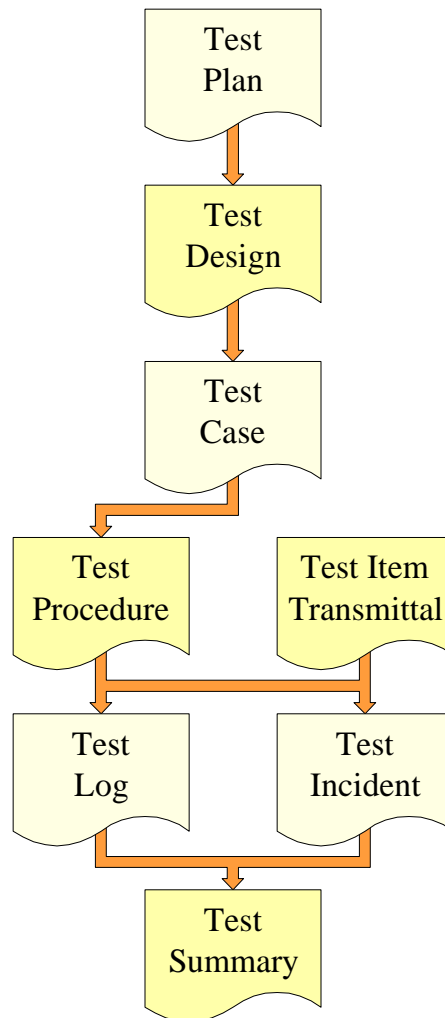
Testing (Post Process)



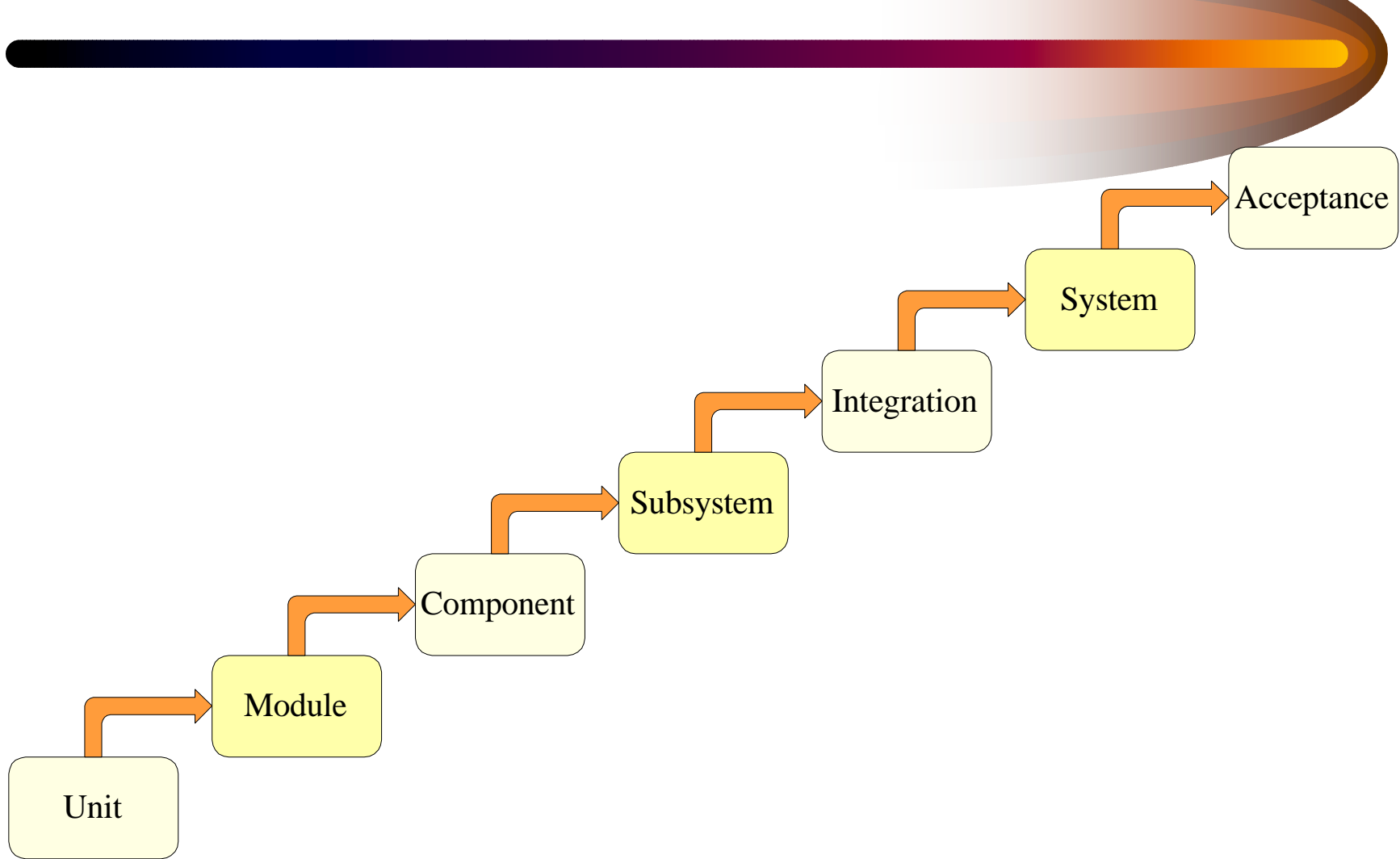
Testing Techniques

White Box (internal)	Basis Path	Flow Graph Notation
		Cyclomatic Complexity
		Deriving Test Cases
		Graph Matrices
	Control Structure	Condition
		Data Flow
Black Box (external)	Specification	Loop
		Graph Based Methods
		Equivalence Partitioning
		Boundary Value Analysis
	Interface	Comparison
		Interface Misuse
		Interface Misunderstanding
		Timing
	Operational	Serialization
		Requirements
Scenario		
		Use

Testing Documents



Testing Levels

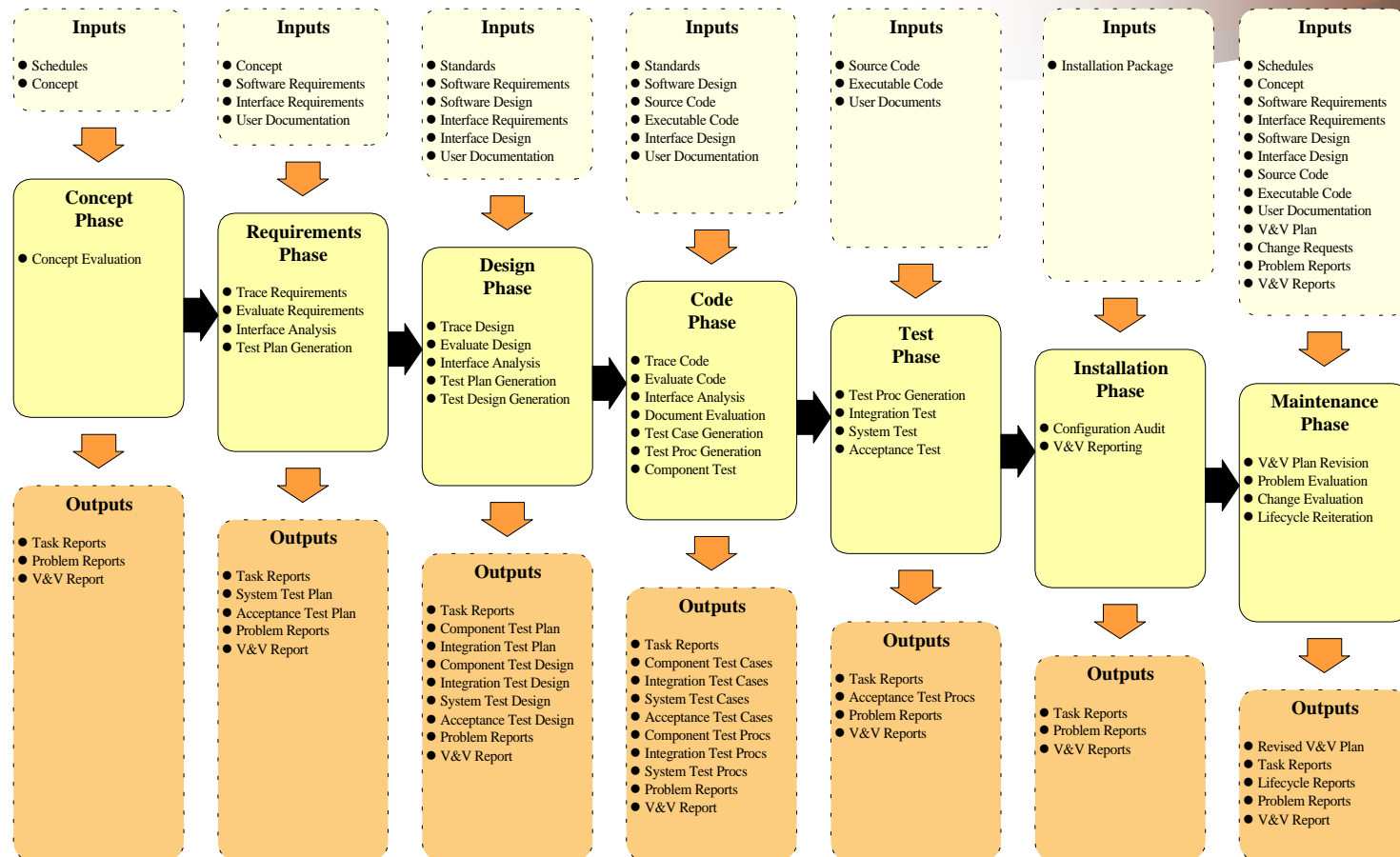


What does Testing Do?

- V&V is the process of determining whether:
 - Requirements for a system or component are complete and correct
 - Products of each development phase fulfill the requirements or conditions imposed by the previous phase

Final systems or components comply with specified requirements

Lifecycle Frameworks (In Process)



“IEEE Standard for Software Verification and Validation Plans,” IEEE Std 1012-1986.

More Lifecycle Framework Tasks

Lifecycle Phase Optional Task	Management	Concept	Requirements	Design	Implementation	Test	Installation	Maintenance
Algorithm Analysis			✓	✓	✓	✓		✓
Audit Performance								
Configuration Control					✓	✓	✓	✓
Functional					✓	✓	✓	✓
In-Process			✓	✓	✓	✓	✓	✓
Physical						✓	✓	✓
Audit Support								
Configuration Control					✓	✓	✓	✓
Functional					✓	✓	✓	✓
In-Process			✓	✓	✓	✓	✓	✓
Physical						✓	✓	✓
Configuration Management	✓	✓	✓	✓	✓	✓	✓	✓
Control Flow Analysis			✓	✓	✓			✓
Database Analysis			✓	✓	✓	✓		✓
Data Flow Analysis			✓	✓	✓			✓
Feasibility Study Evaluation		✓						✓
Installation and Checkout			✓	✓	✓	✓	✓	✓
Performance Monitoring								✓
Qualification Testing			✓	✓	✓	✓	✓	✓
Regression Analysis			✓	✓	✓	✓	✓	✓
Reviews Support								
Operational Readiness							✓	✓
Test Readiness					✓	✓	✓	✓
Simulation Analysis			✓	✓	✓	✓	✓	✓
Sizing and Timing Analysis				✓	✓	✓	✓	✓
Test Certification						✓	✓	✓
Test Evaluation			✓	✓	✓	✓	✓	✓
Test Witnessing						✓	✓	✓
User Document Evaluation		✓	✓	✓	✓	✓	✓	✓
V&V Tool Plan Generation	✓							✓
Walkthroughs								
Design				✓				✓
Requirements			✓					✓
Source Code					✓			✓
Test					✓	✓	✓	✓

“IEEE Standard for Software Verification and Validation Plans,” IEEE Std 1012-1986.

What do Lifecycle Frameworks Do?

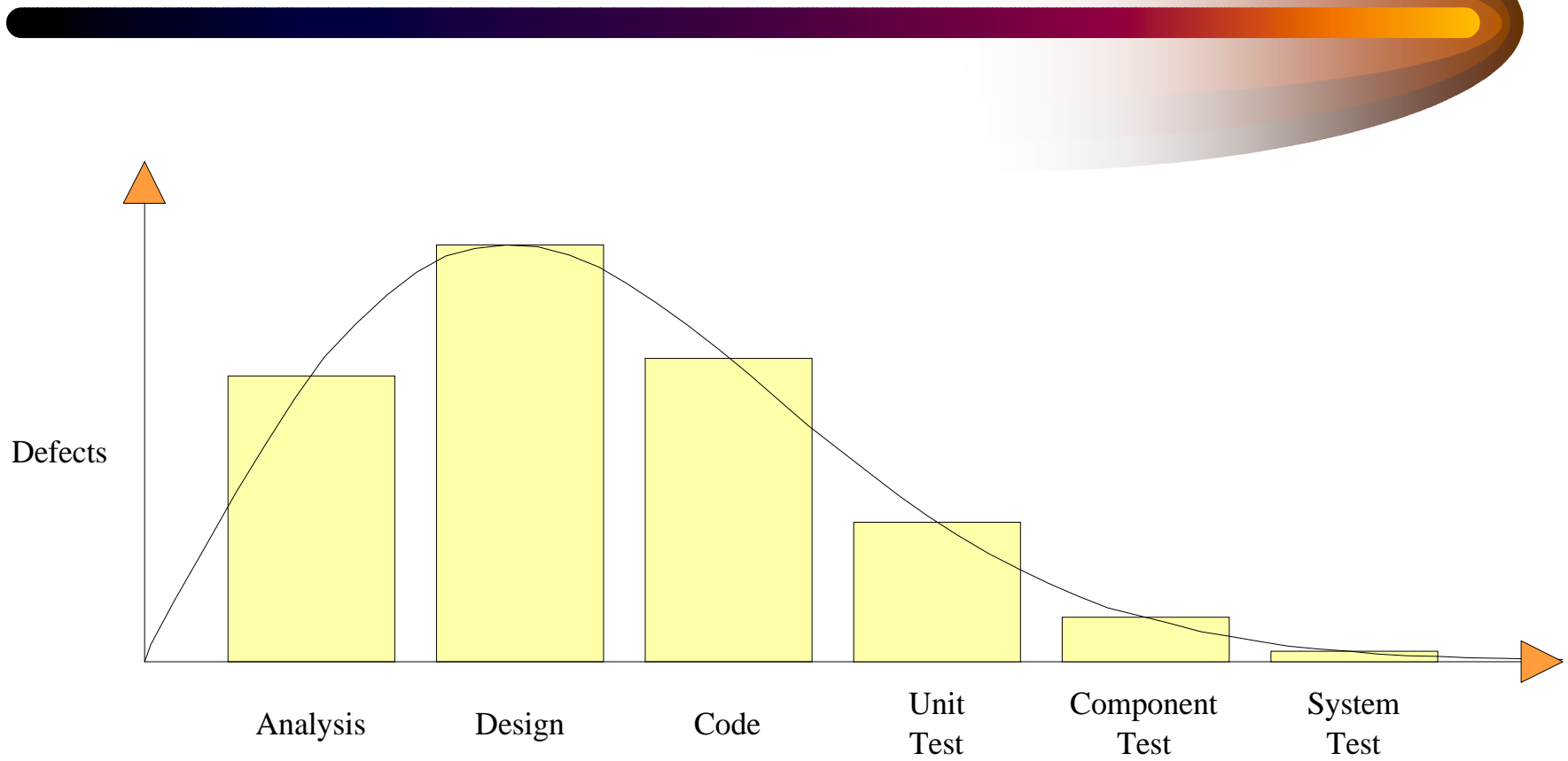


- V&V is the process of determining whether:
 - Requirements for a system or component are complete and correct

Products of each development phase fulfill the requirements or conditions imposed by the previous phase

Final systems or components comply with specified requirements

Lifecycle Methodologies (In Process)



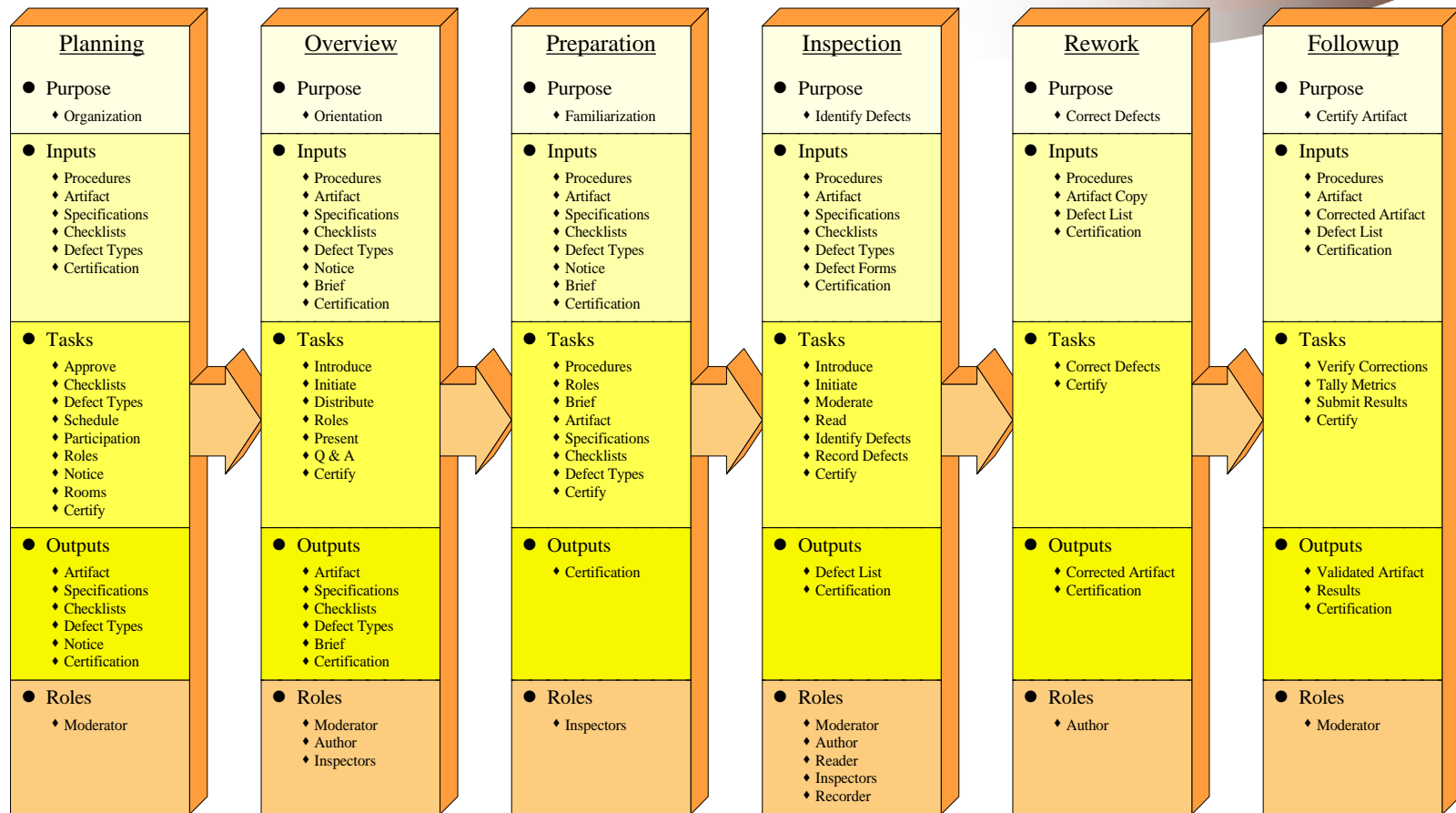
Kan, S. H. (1995). Metrics and models in software quality engineering. Reading, MA: Addison-Wesley.

Basis for Lifecycle Methodology

V&V Activity Lifecycle Phase	System/36	System/38	Ideal	AS/400
Product Objectives	Review	Review	Review	Walkthrough
Architecture	Review	Walkthrough	Inspection	Inspection
Specification	Review	Walkthrough	Inspection	Inspection
High Level Design	Review	Inspection	Inspection	Inspection
Inter Component Interfaces	Review	Inspection	Inspection	Inspection
Low Level Design	Inspection	Inspection	Inspection	Inspection
Code	Inspection	Inspection	Inspection	Inspection
Test Plan	Inspection	Inspection	Inspection	Inspection
Test Cases	Inspection	Inspection	Inspection	Inspection

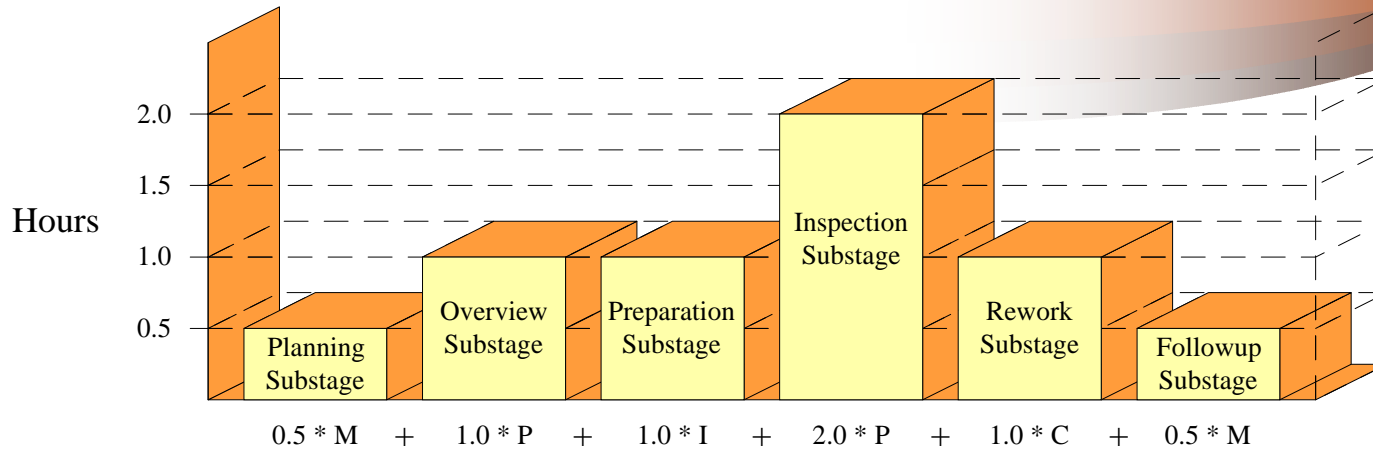
Sulack, R. A., Lindner, R. J., & Dietz, D. N. (1989). A new development rhythm for AS/400 software. *IBM Systems Journal*, 28(3), 386-406.

Software Inspection Process



Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal*, 12(7), 744-751.

Inspection Process Measurability



$$\text{Hours} = \text{Product Size} / (\text{Inspection Rate} * 2) * (\text{Team Size} * 4 + 1)$$

People	10,000 Lines			100,000 Lines			1,000,000 Lines		
	60 SLOC Per Hour	120 SLOC Per Hour	180 SLOC Per Hour	60 SLOC Per Hour	120 SLOC Per Hour	180 SLOC Per Hour	60 SLOC Per Hour	120 SLOC Per Hour	180 SLOC Per Hour
4	1,417	708	472	14,167	7,083	4,722	141,667	70,833	47,222
5	1,750	875	583	17,500	8,750	5,833	175,000	87,500	58,333
6	2,083	1,042	694	20,833	10,417	6,944	208,333	104,167	69,444
7	2,417	1,208	806	24,167	12,083	8,056	241,667	120,833	80,556

Russell, G. W. (1991). Experience with inspection in ultralarge-scale developments. *IEEE Software*, 8(1), 25-31.

Lifecycle Methodology Accuracy

Project	Size	Language	Spec	Design	Code	Unit Test	Comp Test	System Test	First Year	Product Life	Estimate Accuracy
A	680K	Jovial	4	--	13	5	4	2	0.3	0.6	0.6
B	30K	PL/1	2	7	14	9	7	--	3.0	6.0	6.0
C	70K	BAL	6	25	6	3	2	0.5	0.2	0.4	0.3
D	1,700K	Jovial	4	10	15	4	3	3	0.4	0.8	0.9
E	290K	Ada	4	8	13	--	8	0.1	0.3	0.6	0.7
F	70K	--	1	2	4	6	5	0.9	1.1	2.2	2.1
G	540K	Ada	2	5	12	12	4	1.8	0.6	1.2	1.1
H	700K	Ada	6	7	14	3	1	0.4	0.2	0.4	0.4

Kan, S. H. (1995). Metrics and models in software quality engineering. Reading, MA: Addison-Wesley.

What do Lifecycle Methodologies Do?



- V&V is the process of determining whether:

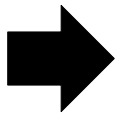
Requirements for a system or component are complete and correct

Products of each development phase fulfill the requirements or conditions imposed by the previous phase

Final systems or components comply with specified requirements

Costs & Benefits

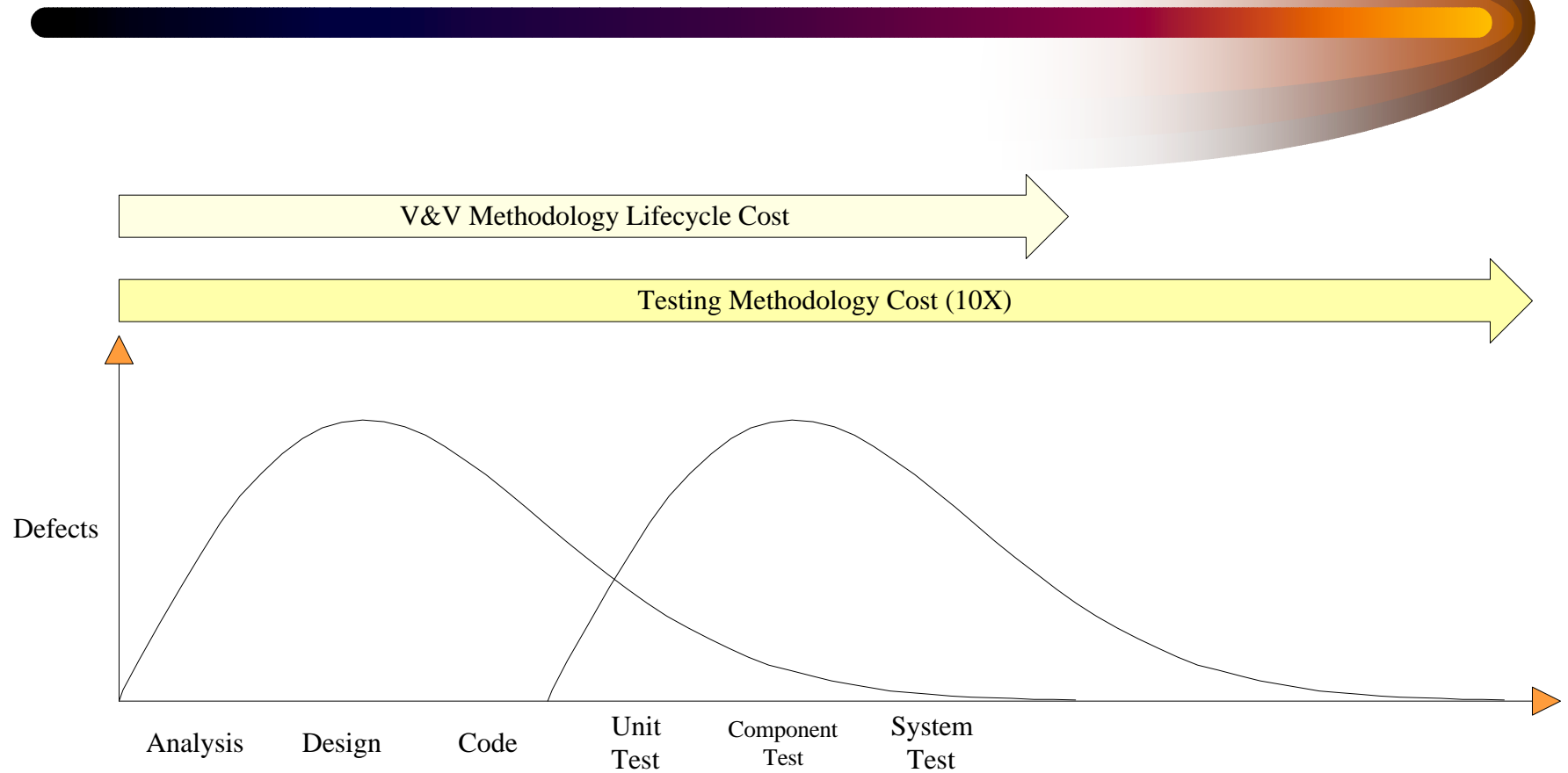
Methodology Cost/Benefits	Lifecycle		Testing	
	PSP sm	Inspection		
Program Size	10,000 SLOC	10,000 SLOC	10,000 SLOC	
Start Defects	1,000	1,000	1,000	
Inspection Hours		708		
End Defects		100		
Inspection Efficiency		1.27 Defects/Hour		
Start Defects		100		1,000
Test Hours		1,144		11,439
End Defects		10		100
Test Efficiency		12.71 Hours/Defect		12.71 Hours/Defect
Total V&V Hours	400 *	1,852	11,439	
Total V&V Defects	1,000	990	900	
Delivered Defects	0	10	100	



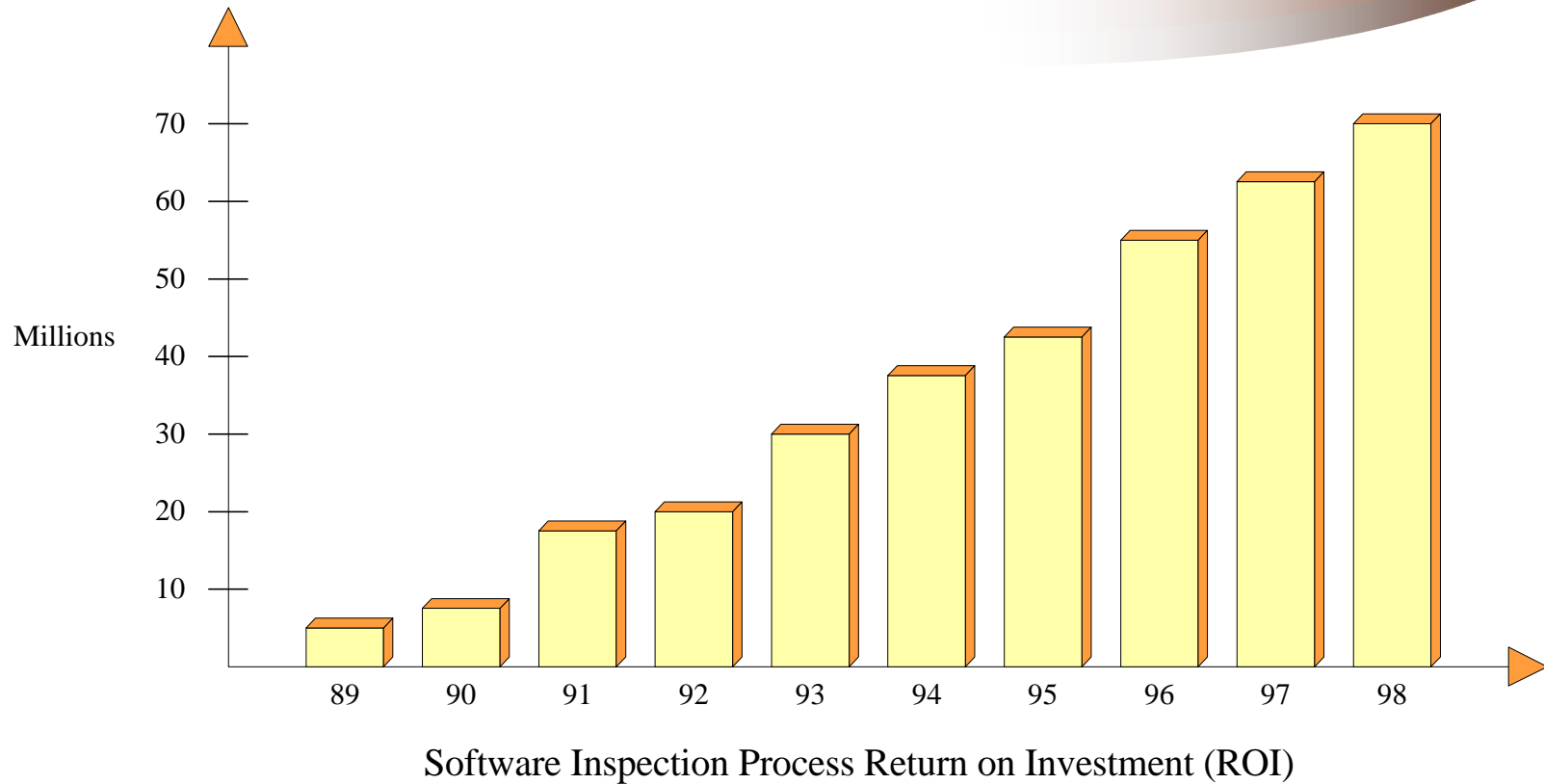
* PSP V&V hours include development time

sm Personal Software Process and PSP are service marks of Carnegie Mellon University.

Costs of Methodologies

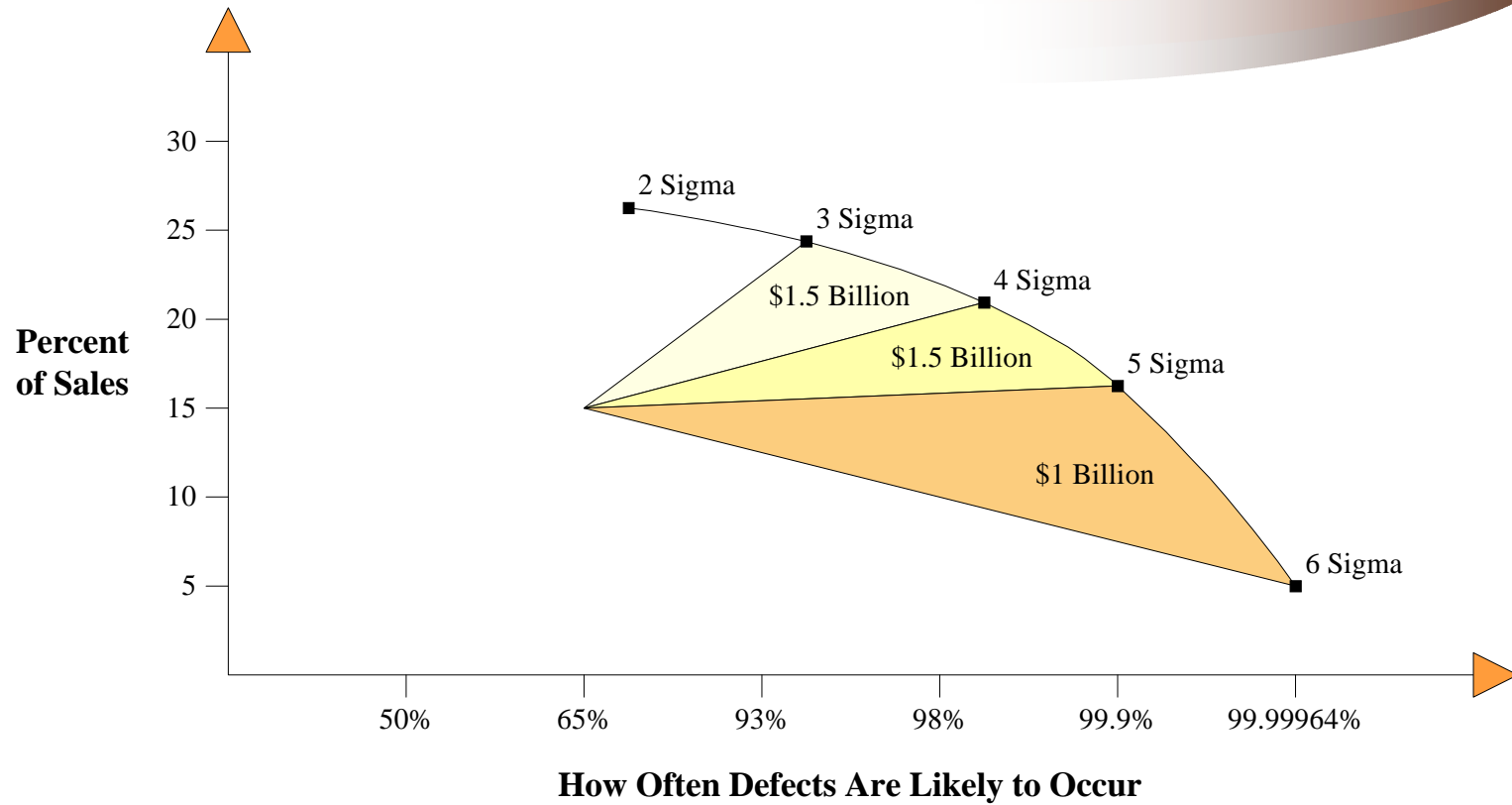


Hewlett Packard



Grady, R. B. (1997). Successful software process improvement. Saddle River, NH: Prentice Hall.

Raytheon



Velocci, A. L. (Nov 1998). High hopes riding on six sigma at raytheon. Aviation Week & Space Technology.

Myths & Misconceptions



- V&V, Quality, and Testing often confused
- Quality and Testing often equated
- Testing and V&V often equated
- Testing believed to be sufficient
- V&V often confused with IV&V
- IV&V believed to be better than Lifecycle Methodologies

Conclusion



- Testing is inefficient and happens too late in the lifecycle
- Lifecycle Frameworks are inundating, non-methodological, and not easily measured
- Lifecycle Methodologies are fast, efficient, measurable, and accurate

Bibliography



- “Modeling and Software Development Quality,” Stephen H. Kan, IBM Systems Journal, Vol 30, No 3, 1991
- “AS/400 Software Quality Management,” Stephen H. Kan, et al., IBM Systems Journal, Vol 33, No 1, 1994
- “A New Development Rhythm for AS/400 Software,” Richard A. Sulack, et al., IBM Systems Journal, Vol 28, No 3, 1989
- “Lessons Learned from Three Years of Inspection Data,” Edward F. Weller, IEEE Software, September 1993
- “IEEE Standard for Software Verification and Validation Plans,” IEEE Std 1012-1986
- “IEEE Guide for Software Verification and Validation Plans,” IEEE Std 1059-1993