The Rapid Deployment of a Defect Analysis Program

by Steven H. Lett

Introduction

In 1997 Lockheed Martin Government Electronic Systems (GES) in Moorestown, New Jersey, was in the midst of a software process improvement initiative. GES develops the radar and combat systems for AEGIS guided missile cruisers and destroyers for both the U. S. Navy and internationally. An organizational goal had been established to improve the GES software processes to the extent that a Software Engineering Institute (SEI) Level 3 maturity rating could be achieved as measured against the SEI Capability Maturity Model (CMM). The GES Software Engineering Process Group (SEPG) was leading the initiative in accordance with a very aggressive schedule. Part of this effort included an upgrade of the current peer review practice of structured design and code walkthroughs to include the more rigorous software inspection methodology, as well as a defect analysis program. Limited resources were available to implement the desired changes. However, in less than two months the software inspection and defect analysis processes were defined, documented, and rolled out to a pilot project, along with a tool set to support the required defect data collection and reporting. Less than a year later, the SEI Level 3 goal was attained with the successful completion of a CMM-Based Assessment for Internal Process Improvement (CBA IPI). The defect analysis program was listed by the CBA IPI assessment team as one of the process strengths exhibited by the software organization.

This paper focuses on the defect analysis aspect of the process improvement task and describes how it was deployed quickly and economically. In particular, the topics covered include:

- The critical steps taken to efficiently define and implement the defect analysis program
- The measurements defined for collection and derivation from the defect removal activities and how they are used
- How simple but effective support tools for automating data collection and analysis were developed
- Lessons learned, including what worked well and what did not

Implementation Process

Defect Analysis Program Goal Establishment

The first steps in implementing a defect analysis program were to establish the purpose and goals of the program and its role in supporting the organization's software process improvement goals. This was essential to set the scope of the task and facilitate decision making during the design and implementation of the program. The primary goals of the program were:

1. To satisfy SEI level 3 CMM criteria, particularly certain key practices within the Peer Review and Software Product Engineering Key Process Areas (KPAs). [1]

2. To set the groundwork for SEI Level 4 by collecting data for assessing process stability and to support the analysis associated with defect removal and defect prevention efforts.

In using the CMM to provide direction in our efforts to improve, it was determined that our peer review and defect analysis procedures must be documented, that training be provided for all involved personnel, and that the following data be collected and analyzed:

- Data on the conduct and results of peer reviews
- Measurements to determine the status of the peer review activities
- Data on defects detected during peer reviews and testing

The value of peer reviews, especially software inspections, in improving product quality, reducing rework, improving productivity, reducing cycle time, and reducing cost is well-documented. [2] [3] Therefore, it is very important to measure the results, status, and defect-removal efficiency of the peer review process and look for opportunities to improve it.

Defect data collected from peer reviews, testing, and operational use provide insight into the quality of the software development processes and the software products that can be used to initiate process improvement.

Process Definition

The next steps in the implementation process were to develop in-house expertise and then define the defect analysis procedure. An essential requirement for this step was bringing in expertise from outside of the organization. This was accomplished in three ways:

1. By bringing in an outside consultant who was an expert in software inspections and defect analysis.
2. By utilizing the documentation made available for sharing within the corporation from other Lockheed Martin businesses.
3. By reviewing some of the voluminous material available on the subjects of software inspections and defect analysis.

The outside consultant conducted training including two software inspection orientations (one for software engineers and one for managers) and a software inspection workshop for the engineers. His training material provided excellent examples of how defect data and software inspection data can be utilized. Other miscellaneous material was readily available and helpful, including books, SEI Technical Reports [4] [5], and information from the Internet.

Utilizing the shared process documentation, including procedural descriptions and guidebooks, from other Lockheed Martin businesses was especially important in expediting this step of research and process definition. Lockheed Martin is a large corporation with a significant number of SEI Level 3, 4, and 5 organizations. Several of these sites have their process documentation available over the corporate intranet. Subsequently, producing detailed procedure descriptions for peer reviews and defect analysis became an editing task to adapt the new procedures to the GES culture and add the best aspects of the other material used.

Measurement Determination

As part of the defect collection and analysis procedure definition, a determination was made of the specific measurements that were needed. These were derived from the defect analysis goals: to measure peer review status, to measure the efficiency of the peer review process, and to collect data on the defects being inserted into the software products to support future analysis. The measurable attributes of these goals were determined. For example, a measurable attribute of peer review effectiveness is the number of defects that escape through a peer review into later development phases, such as a code defect being found in testing. To determine defect leakage such as this, for each defect the development phase where the defect was inserted must be recorded, as well as the defect removal activity (e.g., testing) where the defect was found.

Another aspect of determining measurement requirements was to predict relevant data that could be useful in diagnosing possible causes of process inefficiency. Requiring that peer review preparation time be recorded is an example of this type of measurement. The amount of time individuals spend preparing for a peer review can be assumed to have a direct bearing on the effectiveness of each review. Since it can be anticipated that eventually an analysis will be made as to how the peer review process could be improved, data on preparation time would be considered important information. Therefore, it was included as a measurement requirement.

In choosing the measurements to make in support of the defect analysis goals, the reference material described earlier was used both for guidance and to ensure we used measurements that were in common use within the software industry. This was for two reasons:

- It was felt it would be easier to “sell” the new measurement requirements to management and the software engineers if plenty of examples could be given that much of the industry makes the same measurements.

- Using common measurements would give us the option to use industry data as benchmark data for comparing with our data. This would aid our data analysis and our understanding of what the data may indicate about our defect-removal processes and the quality of our work products.

The required measurements were of two types: collected measurements and derived measurements calculated from the collected data. Table 1 lists the measurement requirements that were defined. The derived measurements are indicated by a mark in the "Calc" column.

Reports

It was decided that three reports would be generated: a Peer Review Report for each individual peer review, a Test Defect Log, and a monthly Project Quality Report. Each of these reports is described below. How these reports are generated is discussed later under "Tool Development."

Peer Review Report - The Peer Review Report is used to document the results of a single peer review. At the conclusion of a peer review, this report is distributed to the peer review participants, the cognizant project manager, and the Software Quality Assurance (SQA) representative for review. The Peer Review Report is comprised of three sections: the Peer Review Record, the Peer Review Defect Log, and the Peer Review Summary.

The Peer Review Record serves as both a form for entering required peer review information and as a report. The information collected and reported in the Peer Review Record includes program, program element, and function identifiers, the work product being reviewed, the type of review (i.e., software inspection or the less rigorous product review), who attended, how much time they spent preparing, how long the review meeting lasted, the disposition of the review, the checklists used, etc. An example of this record is shown in Figure 1.

The Defect Log also serves as both a form for entering the required data for each defect found at a peer review and a record. It includes defect type, defect origin, defect severity, defect category, defect location, time to fix, date closed, and other information. An example of the Defect Log is shown in Figure 2.

The Peer Review Summary is the third component of the Peer Review Report. It provides metrics that profile the conduct and results of a peer review. The information is useful in providing feedback to the product author and the peer review participants about the product and about the types of defects that could be eliminated earlier in the development process. Peer review efficiency and effectiveness measurements are provided to help determine if the peer review was within the normal expected ranges for the particular type of product reviewed. If not, further investigation may be warranted to determine why the peer review was an apparent anomaly. A sample of the Peer Review Summary is depicted in Figure 3. The main sections of the report are:

- General peer review information - At the top of the report is the general information that identifies the date of the peer review and the associated work product.
- Defect Type by Defect Category profile - The Defect Type by Defect Category matrix provides a profile of the defects found during the peer review by Defect Type, Defect Category, and Defect Severity.
- Defect Origin profile - The Defect Origin table in the report plots the major and minor defects found against the phase in which the defects were injected into the product.
- Peer Review efficiency and effectiveness - The table at the bottom of the report provides measures that primarily indicate how efficient the review was for the time invested and how effective it was at finding defects.

Test Defect Log - During unit testing and element integration and testing (EI&T), the software engineers are required to fill out a defect log containing data on each defect they detected. The content of the Test Defect Log is similar to the Peer Review Defect Log described earlier and illustrated in Figure 2.

Project Quality Report - The data from the Peer Review Reports and Test Defect Logs are entered into two project databases: a peer review database and a defect database. A Project Quality Report is issued once a month using the information from these databases. The report is in two forms, as an online report and as a hard copy report. (How the Project Quality Report is generated is described later under "Tool Development"). All project personnel are given access to the report and are encouraged to record their analysis of the data. The peer review/defect analysis process leader from the SEPG reviews the report each month and records an analysis of the data with recommendations of any possible actions to take. Recorded analysis describes trends and anomalies observed in the data. Any subsequent corrective actions and their consequences are also recorded.

Table 1. Measurements for Defect Analysis

Measurement	Calc	Comments	Use
Information on Each Defect Found		From peer reviews & testing	
Change control #		Spec. Change # &/or problem report #	Traceability
Program/Function Info		Program, Element, Version, etc.	Data grouping
Defect type		Wrong, Missing, or Extra	Defect analysis
Defect origin		Phase inserted, e.g., design or code	Product quality; defect leakage; defect removal efficiency
Defect severity		Major or minor	Product quality; defect analysis
Defect category		Documentation, Data, I/O, etc.	Defect analysis
Activity found		Peer review or test type, e.g., Detailed Design Inspection or Unit Testing	Defect leakage; defect removal effectiveness
Defect location		Module, procedure, line #, etc.	Defect closure tracking
Defect description		Concise	Defect closure tracking
Time to Fix		Time taken to fix & reinspect or retest	Total cost assessment; ROI calc
Action item information		Who assigned; when due; when completed	Defect closure tracking
Information on Each Peer Review			
Peer Review Info		Date, product name, product type, reviewers (by role), peer review type, etc.	Data grouping
Disposition		Accepted (completed), Conditional, Re-review	Status tracking
Total Preparation Time		Sum of each participants' time	Calculated measurements
Meeting Time		Length of the meeting	Calculated measurements
# of participants		Sum of % of participation of each participant	To compute total review time
# SLOC reviewed		SLOC = executable Source Lines of Code	Calculated measurements
# Pages of documentation reviewed		Sum of changed pg. portions	Calculated measurements
Total Fix Time		Time it took to fix the defects & reinspect	Total cost assessment; ROI calc
# Major Defects Found	X	Sum of Major Defects found	Peer Review effectiveness
# Major Defects Found by type, category, and phase	X	For Major: Sum of each type, each category, each phase of origin	Defect analysis
# Minor Defects Found	X	Sum of Minor Defects found	Peer Review effectiveness
# Minor Defects Found by type, category, and phase	X	For Minor: Sum of each type, each category, each phase of origin	Defect analysis
Total Defects Found	X	Sum of Major and Minor Defects	Peer Review effectiveness
Total Meeting Time	X	# of participants * Meeting Time	Other calculations
Total Detection Time	X	Total Prep. Time + Total Meeting Time	Other calculations
Total Review Time	X	Total Detection Time + Total Time to Fix	Other calculations
Average Prep. Time per Reviewer	X	Total Prep. Time ÷ # of participants	Prep. Time adequacy
Ave. Prep. Time Review Rate - SLOC/Hr.	X	# SLOC ÷ Ave. Prep. Time per Reviewer	Prep. Time adequacy
Ave. Prep. Time Review Rate - Pgs./Hr.	X	# Pgs. ÷ Ave. Prep. Time per Reviewer	Prep. Time adequacy
Total Peer Review Time per Defect *	X	Total Review Time ÷ Total Defects Found	Peer Review efficiency
Defects Found per Detection Hr. *	X	Total Defects ÷ Total Detection Time	Find time efficiency
Defects Logged Per Hr.	X	Total Defects ÷ Meeting Time	Peer Review efficiency
Defects Found per Page *	X	Total Defects ÷ # Pages	Peer Review effectiveness
Defects Found per KSLOC *	X	Total Defects ÷ (SLOC ÷ 1000)	Peer Review effectiveness
* - also measured for Major Defects			
Peer Review Project Summary Metrics		Accumulated By Project	
All Peer Review measurements listed above		Same as the individual peer review measurements listed above only accumulated for the entire project	See above
# Peer Reviews Completed by Type	X		Peer Review Status
# Peer Reviews In Progress by Type	X		Peer Review Status
Major Defects Per Review	X	Matches available baseline measurement type	Comparison to historical data
Minor Defects Per Review	X	Matches available baseline measurement type	" " " "
Total Defects Per Review	X	Matches available baseline measurement type	" " " "
% of Major Defects Found Per Phase	X	For each phase, Major Defects ÷ Total Major Defects Found in all development phases	Defect removal efficiency

Peer Review Record

Review Title:				Review Date:	
Review ID#:		TOR/SC #:		CPCR#:	
Baseline:		Element:		Change Type:	

Module/Function:		SLOC Size:		Meeting #1 Duration:	
		# Pages Size:		Meeting #2 Duration:	
Errors in this Record:				Meeting #3 Duration:	
				Total (Hrs):	
Product Type:		Review Type:		Life Cycle Phase:	

Checklists Used	
Completeness	
Correctness	
Style	
Rules of Construction	
Multiple Views	
Technology	
Metrics	
AEGIS CPS (Req'd)	

Reviewers			
Role	%	Name	Prep Time
Moderator			
Author			
Reader			
Reviewer			
Reviewer			
Reviewer			
Reviewer			
Total # Reviewers		Total Hours (tenths)	

Information at Review Completion			
Disposition:		Defects Found? (Y/N):	

Comments	
	- indicates cells with formulas, i.e., computed values

Distribution		
Name/Mailstop	Name/Mailstop	Name/Mailstop

Figure 1. Peer Review Record – serves as a data entry form and a printable report

The Project Quality Report includes charts and tables depicting three categories of metrics: *peer review status, product quality, and process efficiency*. The hard copy version contains only project summary metrics combining all program elements. The on-line version provides interactive control for producing the charts and tables for any combination of the project's program versions and program components. The charts and tables in the Project Quality Report include:

- Defect Severity, Category, and Type Profiles
- Defect Analysis by Phase
- Defect Density for Documents and for Code
- Peer Review Status and Process Metrics

Peer Review Defect Log													
CPCR#:					TOR/SC#:				Baseline:			Review Date:	
Element:					Review Code:				Module:			Review ID:	
#	Page	Line	Defect Cat.	Defect Sev.	Defect Type	Defect Origin	Assignee/ Org.	Module or Procedure	Due Date	Date Closed	Hrs to Fix	Defect Description	Response
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
										Total Fix Time		Reviewed By	

Defect Category: Data, Documentation, Interface, Logic, Maintainability, Performance, Standards, Other
Defect Severity: Major, Minor **Defect Type:** Missing, Wrong, Extra **Defect Origin:** Reqs, Design, Code, Unit Test, EI&T, Maintenance

Figure 2. Peer Review Defect Log

The Defect Severity Summary depicts the number of major and minor defects found and fixed in the project's work products for each software development phase, i.e., requirements, design, code, unit test, and EI&T. Figure 4 is an example of this chart in the Project Quality Report. The chart can be used to draw some conclusions about the overall quality of a project's products.

The Defect Category Profile, Figure 5, contains a profile of the defects in each defect category for each phase. This defect profile supports a Pareto analysis for determining the most prevalent sources of defects.

The Project Defect Type Profiles, Figure 6, show the number of wrong, missing, and extra defect types for both major and minor defects by phase. Especially high numbers for a particular type of defect for a particular product, e.g., design documentation, may reveal issues to be addressed. For example, the defect type profile may reveal ambiguity in the requirements if the "missing" or "extra" counts are high in subsequent work products.

The Defect Analysis By Phase chart contains a profile of the injection, removal, and leakage of defects throughout the development life cycle. "Injected defects" equate to the recorded Defect Origin for each defect. "Removed defects" are determined by the peer review or test where they were found. As an example, Figure 7 shows the number of major defects injected and removed during each phase. The chart also depicts the percentage of all major defects removed in each phase. "Escaped" is also plotted for each phase and is the difference between the defects injected and removed, i.e., the defects that escaped the detection process and affect the next activity.

This data provides insight into both process effectiveness and product quality. It is more useful when a software program has completed development and is in use by the customer because a more accurate profile of the developed product's known defects throughout the development cycle can be plotted. After a product has been submitted for customer use, this data should be analyzed to determine which activities are the primary contributors of defects and which have inadequate detection processes. Corrective actions should be taken as a result of the analysis to reduce the number of defects injected and to improve the detection process so that the number of escaping defects is reduced.

The Defect Density For Documents and Defect Density for Code charts depict the density of defects found in each software product. For documents, the number of pages per defect are plotted. For code the defect density is represented as defects per 1000 SLOC (KSLOC). These measurements provide insight into both process effectiveness and product quality. Defect density analysis throughout the development cycle provides a good quality measurement of each product, especially when sufficient historical data on similar products is available for comparison. It can aid in identifying the products and process steps with the most leverage for improvement. Continual comparison against historical defect density data should indicate the effectiveness of the improvement efforts.

Peer Review Summary Report

Review Title:	Sample title	**Review Date:**	1/1/97
Module:	Module A	**CPCR#:**	C12345
Baseline:	B6P1	**TOR/SC #:**	1122A
Element:	CDSIS	**ReviewType:**	DI

	Major Defects				Minor Defects			
Defect Category	**Missing**	**Wrong**	**Extra**	**Total**	**Missing**	**Wrong**	**Extra**	**Total**
Interface	1			1				
Data		1		1				
Logic								
Performance					2			2
Standards								
Documentation						2		2
Maintainability								
Other								
Totals	1	1		2	2	2		4

	Defect Origin					
Defect Severity	**Reqs.**	**Design**	**Code**	**U. Test**	**EI&T**	**Total**
Major	1	1				2
Minor	1	3				4
			Total Defects Found			6

Measurement Name	Value	Comments
# Reviewers	2.5	
# SLOC		
# Pages	25	
Meeting Time (LH)	2.0	Total meeting duration time
Total Prep Time (LH)	4.0	Total time spent preparing
Total Mtg. Time (LH)	5.0	# Reviewers * Mtg. Time
Total Detection Effort (LH)	9.0	Preparation + Meeting Time
Total Fix Time (LH)	1.4	Time to fix defects & reinspect
Total Inspection Time (LH)	10.4	Total time to find and remove
Ave. Prep Time per Reviewer	1.6	Ave. prep time per reviewer
Ave. Prep Time Review Rate - SLOC/HR		Ave. prep time rate per reviewer
Ave. Prep Time Review Rate - Pgs./HR	15.6	Ave. prep time rate per reviewer
Ave. Inspection Time per Defect	1.7	Ave. time to find and remove
Ave. Inspection Time per Major Defect	5.2	Ave. time to find and remove
Ave. Defects Found/Detection Effort Hr.	0.7	Find time efficiency
Ave. Major Defects Found/Detection Effort Hr.	0.2	Find time efficiency
Defects Logged per Hour	3.0	Inspection efficiency
Meeting Review Rate - SLOC/HR		Review rate
Meeting Review Rate - Pgs./HR	12.5	Review rate
Ave. Defects Found per Page	0.2	Inspection effectiveness
Ave. Major Defects Found per Page	0.1	Inspection effectiveness
Ave. Defects Found per KSLOC		Inspection effectiveness
Ave. Major Defects Found per KSLOC		Inspection effectiveness

Figure 3. Sample Peer Review Summary Report

The Peer Review Status in the Project Quality Report reports the disposition status of all peer reviews conducted to date. A table is used to report the total number of peer reviews completed, in progress, or designated for another review for each program element by product type and review type (inspection or review). Peer Reviews are counted as "in progress" if all defects from the review are not yet fixed. A large number of "in progress" peer reviews could mean that there is a backlog of rework being done.

The Peer Review Process Metrics table is the most comprehensive and perhaps most informative part of the Project Quality Report. Table 2 presents a sample of the measurements comprising the Peer Review Process Metrics. Virtually all collected data from peer reviews is represented in the top half of the table. The lower half of the table is comprised of calculated measurements that provide valuable insight into the effectiveness and efficiency of the various types of peer reviews. The data is presented for each product/review type, e.g., code inspection, code review, or design inspection . The bottom of Table 2 contains a description of the product/review type codes used in the header of the table. The Peer Review Metrics data can be analyzed a variety of ways. The following are examples of some of the guidelines that can be used in analyzing the data:

Inspection Time per Defect - The time to find and fix defects in work products should increase over time due to a larger number of products affected by the defects found in the inspected product.

Defects Found per Detection Hour and Defects Logged per Hour - There should be an upward trend in these values as the inspectors' skills improve.

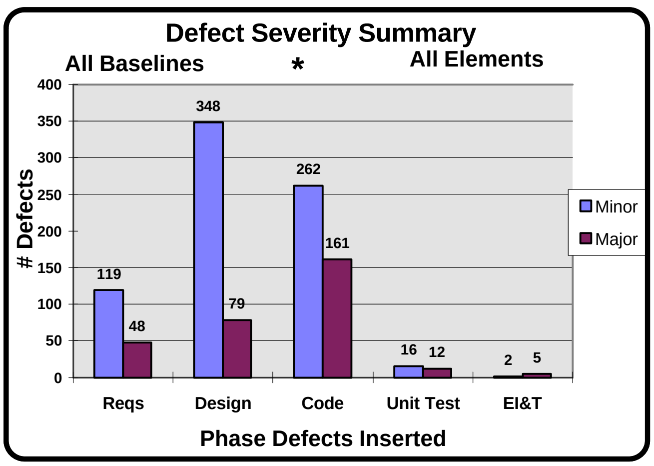

Figure 4. Sample Defect Severity Summary Chart

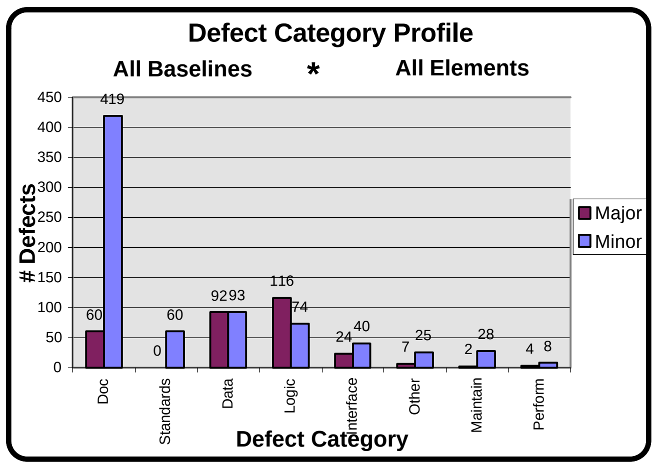

Figure 5. Sample Defect Category Chart

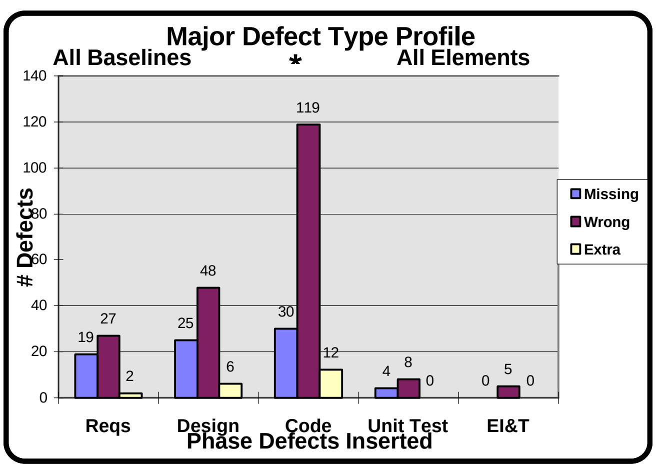

Figure 6. Sample Defect Type Chart

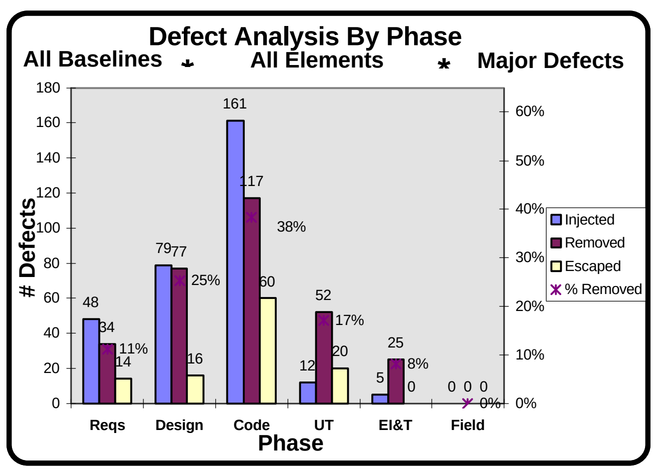

Figure 7. Sample Defect Analysis By Phase Chart

Table 2. Sample Peer Review Process Metrics Table

PEER REVIEW METRICS BY REVIEW TYPE

All Baselines * All Elements

Data	RR	RI	PR	PI	DR	DI	CR	CI	UR	UI	ER	EI	Grand Total
No. of Peer Reviews	50	100	50	100	50	50	40	40	20	20	20	10	956
Source Lines of Code	0	0	0	0	0	0	8800	20000	0	0	0	0	82140
Pages	200	1000	200	600	500	1000			200	200	200	200	6762
No. Reviewers	135	300	100	300	150	180	110	140	44	50	40	25	2738
Preparation Time	40	320	28	200	90	400	120	400	30	45	25	42	1057
Meeting Hours	25	50	22	32	60	70	80	80	30	15	30	15	349
Detection Hours	108	470	72	296	270	652	340	680	96	83	85	80	3231
Time to Fix Hrs	40	40	50	100	50	100	150	220	11	15	2	4	315
Review Hours (Detect. hrs.+Fix hrs	148	510	122	396	320	752	490	900	107	98	87	84	4013
Major Defects	15	100	10	40	22	120	40	150	5	8	3	8	279
Minor Defects	102	250	109	220	180	300	240	360	50	66	4	12	805
Total Defects	**117**	**350**	**119**	**260**	**202**	**420**	**280**	**510**	**55**	**74**	**7**	**20**	**1084**
Ave. No. Reviewers per Review	2.7	3.0	2.0	3.0	3.0	3.6	2.8	3.5	2.2	2.5	2.0	2.5	2.9
Ave. Prep Time per Reviewer	0.3	1.1	0.3	0.7	0.6	2.2	1.1	2.9	0.7	0.9	0.6	1.7	0.4
Ave. Prep Time Rate - SLOC/HR	N/A	N/A	N/A	N/A	N/A	N/A	201.7	175.0	N/A	N/A	N/A	N/A	222.6
Ave. Prep Time Rate - Pgs./HR	13.5	9.4	14.3	9.0	16.7	9.0	N/A	N/A	14.7	11.1	16.0	11.9	18.3
Defects Found/Detection Effort Hr.	1.1	0.7	1.7	0.9	0.7	0.6	0.8	0.8	0.6	0.9	0.1	0.3	0.3
Major Defects/Detect. Hr.	0.1	0.2	0.1	0.1	0.1	0.2	0.1	0.2	0.1	0.1	0.0	0.1	0.1
Defects Logged per Hour	4.7	7.0	5.4	8.1	3.4	6.0	3.5	6.4	1.8	4.9	0.2	1.3	3.1
Meeting Review Rate - SLOC/HR							110.0	250.0					235.4
Meeting Review Rate - Pgs./HR	8.0	20.0	9.1	18.8	8.3	14.3			6.7	13.3	6.7	13.3	19.4
Ave. Defects Found per Page	0.6	0.4	0.6	0.4	0.4	0.4			0.3	0.4	0.0	0.1	0.2
Ave. Major Defects per Page	0.1	0.1	0.1	0.1	0.0	0.1			0.0	0.0	0.0	0.0	0.0
Ave. Defects Found per KLOC							31.8	25.5					13.2
Ave. Major Defects per KSLOC							4.5	7.5					3.4
Ave. Defects/Review	2.3	3.5	2.4	2.6	4.0	8.4	7.0	12.8	2.8	3.7	0.4	2.0	1.1
Ave. Major Defects/Review	0.3	1.0	0.2	0.4	0.4	2.4	1.0	3.8	0.3	0.4	0.2	0.8	0.3
Review Time per Defect	1.3	1.5	1.0	1.5	1.6	1.8	1.8	1.8	1.9	1.3	12.4	4.2	3.7
Review Time per Major Defect	9.8	5.1	12.2	9.9	14.5	6.3	12.3	6.0	21.4	12.2	29.0	10.4	14.4

Peer Review Code = XY where		
X = Product		**Y = Review Type**
R = Requirements	U = Unit Test Procs.	R = Product Review
P = Prelim. Design	E = EI&T Procs.	I = Software Inspection
D = Detailed Design	O = Other	
C = Code		

Meeting Review Rate - This measurement should be used in conjunction with the other effectiveness metrics to determine if peer review meetings are covering the review material at an effective speed. Slow rates may be caused by unprepared participants or too much discussion taking place. Rates that are too fast may result in poor effectiveness in finding errors. Over time the optimum meeting review rates should be determined for the project.

Average Defects Found per Product Size - The higher the rates, especially in the earlier software development activities, the better the final product quality should be. If rates are low, the product may be of extremely high quality (usually when the software process is mature), or else the inspectors need to be more thorough.

Average Defects Found per Review - This metric is included because it can be compared to historical design and code review measurements on the AEGIS projects to determine if improvement from past peer review practice has occurred. This comparison is valid, however, only if the average product size per review is the same then as now.

Tool Development

Once the defect analysis procedure, the required measurements, and reports were defined, it was necessary to consider how the data would be collected, processed, and reported. Obviously, some type of software tools would be needed to collect and store the data, and to generate the reports needed for review and analysis. With limited time and resources, it was decided to start with simple support tools. It was thought that there would be plenty of time later to evolve and enhance the tools when we were more knowledgeable about how the whole process could be improved. A Microsoft® (MS) Excel spreadsheet system was developed for collecting and reporting the peer review and defect analysis data. This was done with the idea that the spreadsheet system initially implemented would serve as a prototype until there was time to develop a more sophisticated system. This proved to be a key factor in expediting the development of the defect analysis program for a number of reasons:

- MS Excel could very easily generate the required charts and tables needed for data analysis.
- Simple two-dimensional database structures in Excel were sufficient to support the database requirements for storing and retrieving the peer review and defect data.
- MS Excel expertise was more readily available among the defect analysis process personnel than any other type of database expertise. Also, the use of MS Excel was very widespread among the target users of the defect analysis program.
- As new tools are developed in the future, any data stored in MS Excel databases could most likely be very easily exported to another database application.
- MS Excel tools can be continually enhanced and automated through the creation of macros.

The key components of the spreadsheet system are three Excel workbook files, each comprised of multiple spreadsheets. The workbook files are the Peer Review Report, the project's Peer Review Database, and the project's Defect Database. Figure 8 illustrates the relationship between these files within the data collection and reporting process.

The Peer Review Report contains the Peer Review Record, Defect Log, and Peer Review Summary worksheets. These worksheets were described earlier in the "Reports" section of this paper and are illustrated in Figures 1, 2, and 3, respectively. The Peer Review Report file is a template file designed to be used for direct data entry during the peer review (assuming a Personal Computer (PC) is available in the meeting room). However, forms for handwritten entry are available and are forwarded to data entry personnel for creating a Peer Review Report file.

The Peer Review Report workbook also contains worksheets that organize the specific data to be transferred to the two databases in the database format, i.e., one row of data per data record. This facilitates the transfer process. Also contained in the file are a number of macros to facilitate data entry and printing, to provide instructional help, and to audit the report for data entry errors or omissions. (Not all of these features existed initially).

The Peer Review Database MS Excel workbook is the repository of peer review data transferred from each Peer Review Report. One worksheet in the workbook contains the database. Each row in the database represents a single record for each peer review. Most of the database fields are described in Table 1 under "Information on Each Peer Review," (except for the calculated averages and ratios listed and the defect type and category information). Spreadsheets are included in the workbook for generating the Peer Review Status and Peer Review Process Metrics (Table 2) reports contained within the Project Quality Report described earlier.

A key capability of MS Excel utilized extensively for generating the charts in the Project Quality Report is the pivot table function. Pivot tables are based on the database worksheet in each database workbook and allow subsets of the data to be grouped in small tables for direct viewing, such as shown in Table 2, or for use in generating charts, such as shown in figures 4 through 7. Pivot tables can be designed to be interactive by allowing any database fields to be set up as "Page" selectors for viewing subsets of the pivot table data. For example, you'll notice that Table 2 has "All Baselines" (i.e., AEGIS program versions), and "All Elements" (i.e., major program elements) in the header. Direct interaction with the pivot tables in the Peer Review Database file allow selection of combinations of any baseline with any program element for more selective viewing of the data.

Macros are utilized within the Peer Review Database file for transferring data from the Peer Review Report files into the database, for controlling the pivot table page settings on all the pivot tables in the workbook at the same time, and for printing the reports.

The Defect Database workbook is the repository of the individual defect data copied from each Peer Review Report and Test Defect Log. One worksheet in the workbook contains the defect database. Each row in the database represents a single defect from either a peer review or test. The database fields are primarily those listed in Table 1 under "Information of Each Defect Found." Pivot table worksheets and charts are included in this workbook for all of the defect profile reports within the Project Quality Report as described earlier (see figures 4 through 7). Macros are also included for the same purposes as for the Peer Review Database.

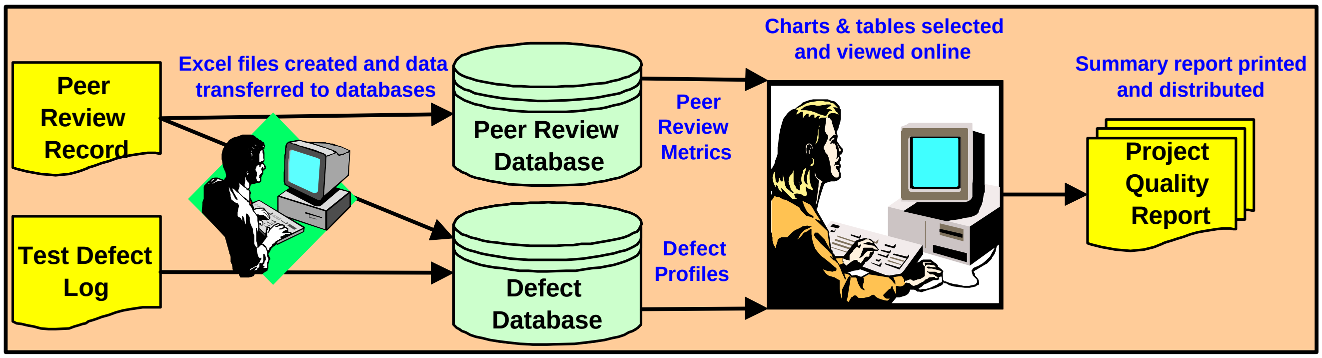

Figure 8. Data Collection and Reporting System

Training

Once the defect analysis goals were established, the procedure defined, the measurement and reporting requirements formulated, and the support tools developed, it was time to roll out the new software inspection process and defect analysis program to the managers and engineers on the selected pilot project. Software inspection methodology training had been performed earlier by a consultant. Therefore, a single orientation course was developed that included both the new peer review process and the defect analysis program. The new written procedures for peer reviews and defect analysis were addressed in the training. The rationale and use for each new measurement was explained, as was each aspect of the Peer Review and Project Quality Reports and the system of Excel spreadsheets. At the completion of the training, direction was given to begin working in accordance with the new procedures.

Follow-up and Process Improvement

Several months after the new procedures were rolled out, a focus group was formed to identify issues and aspects of the process that could be improved. The focus group consisted of software engineers from each program team. The process champion who established the new procedures facilitated the meetings.

The engineers had one issue of primary importance. That was the annoyance caused by the additional paperwork they were required to fill out. Very few of the engineers were entering the data directly into the spreadsheets during the peer reviews. None of the workrooms used for peer reviews were equipped with PCs, so a laptop computer would have had to be checked out for each review. The engineers opted instead to use the handwritten forms. This meant a data entry person would then enter their peer review data into the Excel files described earlier. This caused a delay until the cognizant engineer received a hard-copy version of their peer review report.

This problem was addressed by first upgrading the Peer Review Report file to be more user friendly. Macros were added to provide help descriptions, to automate many of the entries, and to audit the worksheets for errors or omissions (see Figure 1). Hands-on training was then given to all potential peer review recorders to ensure they were comfortable with using the Excel spreadsheets for entering their data. In addition, a PC was installed in the main workroom used for peer reviews, and a common file server was established for storing the peer review and defect data files. (At the current time work has started to create a Microsoft Access front-end for all data entry. This would remove the requirement to maintain separate Excel files for each peer review).

Another major complaint from the engineers was the filling out of multiple forms for a small change. For example, a single source line code change to fix a problem required completing separate peer review forms for any design change, coding change, unit test procedure, or EI&T procedure changes, even though they were all reviewed in one meeting. In response to this, provisions were made to treat small problem fixes as a single product package.

Summary

The defect analysis program, in conjunction with an upgrade of the peer review process to include software inspections, was implemented at Lockheed Martin GES in the relatively short period of time of two months. The goals for the defect analysis program were successfully met. SEI level 3 criteria for peer reviews and defect analysis were satisfied and a good baseline of data was established for SEI level 4. More importantly, however, the data has provided improved insight into the effectiveness and efficiency of our defect-removal activities. The need for improvement is now apparent and the data has helped focus our process improvement efforts.

The critical factors that lead to the successful, rapid deployment of the process changes, were as follows:

1. Starting with clearly understood goals to focus the effort and prevent rework. Using the SEI's CMM as a framework for our software process improvement also provided direction and focus.
2. Utilizing as many resources and as much expertise from outside the organization as possible, including consultants, conference material, technical reports, books, etc. The most significant gain was achieved by utilizing written procedures and guidebooks made available from other sites in the corporation – a true sharing of "best practices." By purposefully utilizing standard measurements from across the software industry, time was not wasted trying to "reinvent the wheel."
3. Starting with simple tools at first and then improving them after living through the new process for awhile. If we had tried to develop or procure more elaborate tools, the implementation would have taken much longer.
4. Taking the time to prepare and deliver training for all personnel affected by the process changes was well worthwhile. Misunderstandings and errors did occur, but not like it would have been if the training wasn't given. Also, more training proved to be essential in responding to process problems identified by the focus group.
5. Continual monitoring and follow-up proved to be essential in correcting early mistakes and misunderstandings. Also, unanticipated areas of awkwardness in the new process needed to be addressed before they resulted in eventual process breakdowns. The focus group concept was very helpful in addressing the engineers' primary concerns.

A number of lessons were learned from this experience. We had underestimated the time required, after process roll out, to monitor the process, and address process inefficiencies and misunderstandings. We learned that many aspects of a new process must be reiterated until it is apparent the engineering staff has internalized it. We learned that engineers' concerns need to be addressed and a continual effort made to improve automation of data collection tasks. When tasks are automated, hands-on training is needed to institutionalize the use of new tools. We also underestimated the time needed to respond to the opportunities for improvement indicated by the results of our defect data analysis. Since our organization is committed to achieving SEI Levels 4 and 5, this problem will be addressed. Plans are in place to develop the infrastructure, knowledge, and cultural mindset for continual process improvement.

References

1. Paulk, Mark C., Weber, Charles A., Chrissis, Mary Beth, et al., *The Capability Maturity Model: Guidelines for Improving the Software Process.* Reading, Mass., Addison-Wesley Publishing, 1995.
2. Gilb, Tom, Graham, Dorothy, *Software Inspection*. Reading, Mass., Addison-Wesley Publishing, 1993.
3. Humphrey, Watts S., *Managing the Software Process*. Reading, Mass., Addison-Wesley Publishing, 1990.
4. Florac, William A.., *Software Quality Measurement: A Framework for Counting Problems and Defects* (CMU/SEI-92-TR-22). Pittsburgh, Pa., Software Engineering Institute, Carnegie Mellon University, September 1992.
5. Baumert, John H., McWhinney, Mark S., *Software Measurements and the Capability Maturity Model* (CMU/SEI-92-TR-25). Pittsburgh, Pa., Software Engineering Institute, Carnegie Mellon University, September 1992.

Steve Lett

Mr. Lett has twenty-23 of software engineering experience as a software engineer and as a project manager developing real-time Command and Control software applications, primarily on the U.S. Navy's P-3C Anti-Submarine Warfare program and the AEGIS guided missile cruiser and destroyer programs. He has extensive experience in leading process improvement initiatives, especially in the areas of project management, software engineering processes, software metrics, training, group problem-solving, and employee empowerment. He presented a paper, "An Earned Value Tracking System for Self-Directed Software Teams," at the recent 1998 (U.S.) SEPG and European SEPG Conferences. Since early 1997 he has been a full-time member of the Lockheed Martin Government Electronic Systems (GES) Software Engineering Process Group (SEPG).