An Analysis of the Correlation between System Engineering Defect Phase Containment and System Engineering Hours at General Dynamics C4 Systems

By

Jennifer Fried

Advisor

Hessam S. Sarjoughian
Arizona Center for Integrative Modeling & Simulation
Computer Science and Engineering Department

An Applied Project Presented in Partial Fulfillment of the Requirements for the Degree Master of Computer Science

ARIZONA STATE UNIVERSITY

May 2005

Instructions.

REPORT OF FINAL MASTER'S WRITTEN OR ORAL EXAMINATION
ARIZONA STATE UNIVERSITY
DIVISION OF GRADUATE STUDIES

1. Part I. The student completes Part I and submits the Report to the Academic Unit following all academic unit deadlines and procedures. The student's Program of Study must be approved by the Division of Graduate Studies before the examinations ,are scheduled.
2. Part II. After the examination, the examining committee members complete Part II, providing the date of the examination, signing the form, and indicating their votes
of Passed or Failed.
3. Part III. The he,ad of the academic unit completes Part III
noting the final result of the examination(s). The final result is based on the majority vote of the committee members.

4.

5.

Notification. The academic unit sends the student a written statement of the results of the examination.
Submission. The completed Report should be submitted immediately to the Division of Graduate Studies, Wilson Hall Lobby, mail code 1003.
All test results, including failures, should be reported. Failure in an examination is considered final unless the student petitions for a re-examination, the supervisory committee and the head of the academic unit recommend, and the Division of Graduate Studies Dean approves a retake.

PART I. Student Information. To be completed by the student. Please print or type.

NAME OF STUDENT (Last name, first name, middle initial): Fried Jennifer A.	ASU I.D. NO: 600-56-7851
ADDRESS (NO., STREET, APT.): 51 W. Windsor Ave.	TELEPHONE: 602-245-9929
CITY, STATE, ZIP: Phoenix, AZ 85003	
MASTER OF: Computer Science	MAJOR:
SIGNATURE: Jennifer Fried	DATE: 5-6-05

Part II. Examination Date and Result. To be completed by the examining committee.

TYPE OF EXAMINATION: ☐ ORAL ☒ WRITTEN		
DATE OF EXAMINATION: 5-6-05	TIME:	PLACE:

CHAIR OR CO-CHAIR
Dr. Hessam Sarjoughian

ABSTRACT

General Dynamics C4 Systems (GDC4S) is a government contractor that employs approximately 9300 people worldwide. As an independently evaluated SEI Level 5 and ISO 9001 compliant company, GDC4S collects many productivity, performance, effectiveness, and quality metrics on the system software that it produces. The company is working to position itself as a leading system integrator and as such, system engineering is gaining focus. GDC4S is lacking useful system engineering metrics to help evaluate system engineering in a similar way to software. As a result, the engineering leadership team has proposed several new system engineering metrics. One such metric emphasizes the importance of system engineering by plotting system engineering hours on a project as a percentage of its total personnel hours against the percent cost and schedule variation over time. This metric is based on data from the System Engineering Center of Excellence (SECOE) project 01-03 presented at the 2004 International Council on System Engineering (INCOSE) International Symposium, in which 42 projects are being evaluated on an ongoing basis from 2001 onward.

In this project, the proposed system engineering metric is extended to evaluate system engineering product quality. General Dynamics C4 Systems uses a metric called CRUD (Customer Reported Unique Defects) to measure software quality. However, this metric is not used specifically for system engineering. Two GDC4S projects that are similar in size and type (system engineering makeup) are analyzed by showing system engineering hours as a percentage of total engineering hours against system – specific CRUD over time. In addition, defect Phase Containment and Phase Differential is analyzed for this CRUD, as it is for software engineering, to illustrate the need for the new system engineering CRUD metric.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION AND COMPANY BACKGROUND

General Dynamics is a company operated as several separate business units, each of which dictates its own policies and procedures. General Dynamics C4 Systems is one such business unit. At General Dynamics C4 Systems, there is no overall company policy for collecting metrics; each of the four divisions collects the metrics that are useful for analyzing its specific business. One exception to this policy is the President's Metrics, which are rolled up at the division level and presented to the President of GDC4S quarterly. These are On-Time Delivery Performance (OTD), Acceptance Test Rate (ATR), Software CMMI Assessment Level, and Systems Assessment Level (SE-CMM). The four metrics from each division are then used to represent GDC4S performance and provided to the General Dynamics corporate office.

In addition to the President's Metrics, the engineering leadership team for each division, which has representatives from all engineering disciplines, makes the decision as to which metrics are appropriate for its own division. The required metrics are documented in the division-level development plan for each discipline. Each project is required to adopt these development plans, and any tailoring is approved by quality assurance. Metrics are developed in large part to support the CMMI Level 5 goals of continuous process improvement and root cause analysis. A typical list of *software metrics* can be found in Appendix B, although they are different for each division.

A list of typical *system engineering metrics*, from the same division as the one in Appendix B, can be found in Appendix C. Most of these metrics are very similar or exactly the same in both disciplines, aside from the ones that are specific to either system or software engineering. However, there are three metrics that seem to be notably absent from the system engineering metrics list. These are Phase Introduced/Discovered, Phase Containment, and CRUD, and they are circled in red in Appendix B.

GDC4S is becoming increasingly involved in system engineering work in its role as system integrator for large-scale defense projects. Because of this, the engineering leadership team is interested in creating new, meaningful metrics to help better manage system engineering effort. One metric that has been created by this team plots system engineering hours on a project as a percentage of its total personnel hours against the percent cost and schedule variation over time. The hypothesis is that there is an optimal percentage of system engineering on a project where cost and schedule variance is minimized.

The intent of this project is to extend the idea of this metric to system engineering quality and effectiveness, both of which types of metrics are largely absent from the system engineering metric list in Appendix C. In Appendix B, the three circled metrics that are not implemented for system engineering relate to effectiveness and quality. Phase Introduced and Discovered, heretofore referred by its common name Phase Distribution, analyzes defects in order to see what phase in the engineering lifecycle they are most often introduced in and discovered in. Trends in the data may indicate issues; for example, finding most defects very late in the cycle could indicate ineffective peer and phase-end reviews. Phase Containment analyzes the differential between the phase in the engineering lifecycle a defect is introduced in, and how many phases later it is discovered. As defects are more costly to fix as they are found later in the cycle, it is desirable to minimize the time between introduction and discovery, i.e. have a differential equal to zero (total phase containment). A lengthy average time between introduction and discovery of defects would also indicate ineffective peer and phase-end reviews, as a defect could go undetected for many phases, when most defects should ideally be found in the same phase in which it was introduced.

The CRUD metric examines defects that have been already found in order to predict how many undiscovered defects are left in a product after a given amount of testing is performed. This project uses CRUD in a different way; instead of predicting the amount of undiscovered defects left in a product, CRUD is plotted against system engineering hours over time in order to show a correlation between system engineering hours spent on a project and product quality.[1] The proposed system engineering metric used as a basis for this project indicates a relationship between system engineering hours and cost and schedule overrun on a project such that there is a percentage of system engineering hours where cost and schedule overrun are minimized. The new system

[1] Product quality in this case is defined as a product with a low number of Customer Reported Unique Defects.

engineering CRUD metric that will be examined in this project indicates that there is a similar relationship between system engineering hours spent on a project and its product quality.

CASE STUDY: DIGITAL MODULAR RADIO AND RESCUE21 PROJECTS

In order to analyze these new metrics on system engineering, it was necessary to select two projects from GDC4S that are similar in terms of engineering makeup, with the goal being to select two projects that are primarily systems/software with very little hardware. The projects selected are Digital Modular Radio and Rescue 21.

Background of Projects Selected for Study

Digital Modular Radio (DMR)

Digital Modular Radio (DMR) is a next-generation Software Defined Radio (SDR) used by the U.S. Navy. This project is in the Communication Networks Division (CND) headquartered in Scottsdale, Arizona. As part of the DMR program, GDC4S is delivering platforms capable of interoperability with disparate tactical systems. These systems are fully programmable and include embedded software-programmable cryptography.

The DMR contract was awarded in 1998, and the first incremental delivery was in 1999. It is an ongoing project with an average cycle time of 9 months. As the primary work on DMR consists of the development of software waveforms on a relatively stable hardware platform, the majority of engineering hours is spent on software. The percentage of hours spent on system engineering is also relatively high, with very few hours spent on hardware. The composition of the DMR project by engineering hours spent for each engineering discipline is found in Figure 1.

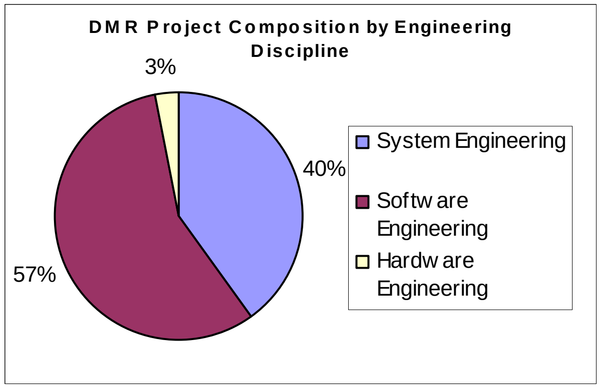

Figure 1: DMR Project Composition

Rescue 21

Rescue 21 is a project in the Space and National Systems Division (SNSD) headquartered in Scottsdale, Arizona. Rescue 21 is a maritime emergency system developed by General Dynamics C4 Systems for the U.S. Coast Guard that will replace its outdated national distress communications system. The National Distress System is the radio system that mariners use to communicate with the Coast Guard in emergencies. The Rescue 21 system will be the nation's primary maritime emergency system for the more than 78 million boaters and 13 million vessels that navigate coastal and intercoastal waters. The system will greatly improve the Coast Guard's ability to detect mayday calls from boaters, pinpoint the location of the source of the call, and coordinate rescue operations.

The Rescue 21 contract was awarded in 2002, and the first incremental delivery was in 2004. This is an ongoing project with increments averaging 12 months. This project consists primarily of integrating existing systems and Commercial Off The Shelf (COTS) software. Therefore, the percentage of system engineering hours is high in comparison to the hours for other applicable engineering disciplines. The composition of the Rescue 21 project by engineering hours spent for each engineering discipline is found in Figure 2.[2] A full description of each type of engineering found in Figures 1 and 2 can be found in Appendix D.

[2] Note that the "other engineering" category is specific to the way that Rescue 21 performs time charging and therefore is not present in the DMR project.

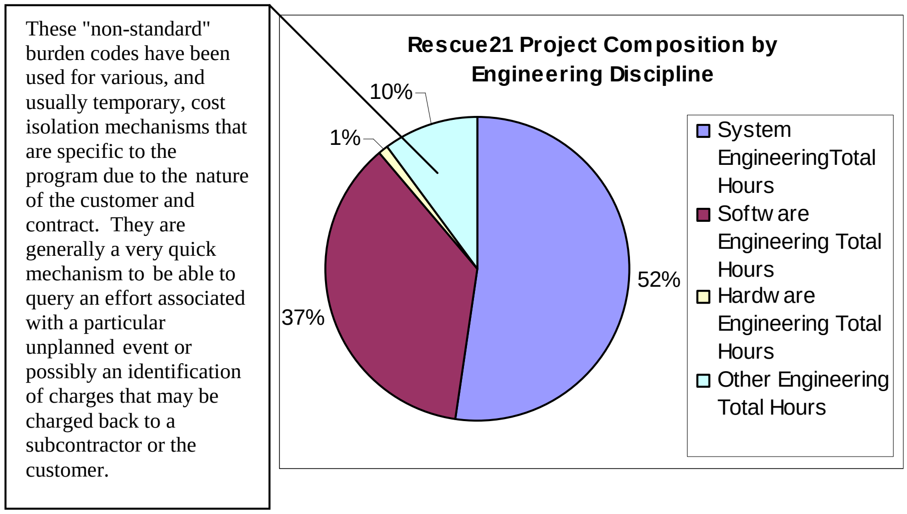

Figure 2: Rescue 21 Project Compostion

Methods used for Study

DMR and Rescue 21 both use the same tools to document defects and employee hours; however, they use them slightly differently. Both use Rational Clearquest as a defect tracking and management tool. Every defect written against the product, both before and after its release, is tracked in the database as a Process Change Request (PCR). PCRs move through the engineering lifecycle in Clearquest, eventually ending up in the closed state when they are fixed. Every PCR has certain elements associated with it, including a unique ID used to track it in the database, the date it was opened, a headline (one line description), the phase in which it was discovered, and the phase in which analysis has shown it to have been introduced. There are many other elements to a PCR, but they are irrelevant for this project's purpose. A sample of the data collected from DMR can be found in Appendix E, and a sample of the data collected from Rescue 21 in Appendix F.

It is important to note that, because the project focuses on system engineering work products, only PCRs with a phase introduced of 1, 2, or 3 are examined. These correspond to the system engineering phases in the engineering life cycle, which are system analysis, system level requirements, and preliminary design; therefore, defects with a phase introduced of 1-3 are system engineering work product defects. For the purposes of the CRUD metric, only *customer reported* defects were examined. Customer reported defects are considered by GDC4S to be those that are discovered out of phase. For example, a defect that was introduced in the requirements phase and found in any later phase from preliminary design to maintenance is considered customer reported.

Both projects also use an in-house software system called Gtime to track all labor hours. Each project is assigned one or labor charge numbers, which are used to track time. These numbers are different for each project, and each project may only have one number that denotes the entire project or several different numbers that split the work into different work products. These numbers are assigned at the beginning of the project, and they are listed in the project's Work Breakdown Structure (WBS). This portion of labor charging helps to track what work product people are working on, at a high level. In order to get more granularity, burden codes are also used in Gtime. Burden codes always consist of two to three characters. They are used differently in every project, but one character always denotes the discipline, such as software engineering or system engineering. The other mandatory character designates the subprocess the worker is participating in, like generating, evaluating, reworking, supporting, or testing. An optional character may specify the engineering lifecycle phase. The burden area is another field in Gtime that specifies only the subprocess from the burden code. There are other pieces of data collected by Gtime, but they are irrelevant to this project and will not be discussed. A sample of the labor hour data from DMR can be found in Appendix G, and a sample of the data collected from Rescue 21 in Appendix H. For this project, data was extracted from both of these sources for the longest time period for which data was available for both DMR and Rescue 21, which was about two years, beginning September of 2002 (when the Rescue 21 project began).

Results: Phase Distribution Metric

The first data examined was the PCR data, approximately 1200 records for both projects. This data was then used to analyze Phase Distribution and Phase Containment for each project. In order to compute Phase Distribution, the number of PCRs introduced in each of the phases considered (1-3) was summed, as was the number of PCRs discovered in each of the latter phases (4-8). The results for the DMR project appear in graphical form in Figure 3, and those for Rescue 21 in Figure 4.

Note that DMR and Rescue 21 have different levels of granularity in their PCR collection. DMR uses the engineering phases most typical for projects at GDC4S, which are 1=system analysis (planning), 2=requirements analysis, 3=preliminary design (system design), 4=detailed design (software design), 5=code and unit test, 6=integration and test, 7=system test, and 8=maintenance (customer released). Rescue 21 however, opts for a finer granularity in their PCR data collection, with additional phases 9 through 12 representing specific types of integration and system testing. Although there are different types of integration test, they are all considered to be the same phase. Similarly, although there are different types of system test, they are all considered to be the same phase. Therefore, for the purposes of this project, all integration test types are grouped into phase 6, and all system test types are grouped into phase 7.

In looking at the data, both projects discover most system engineering defects through the integration and test process. For the majority of the PCRs the phase difference between introduced and discovered appears to be 4 to 5 phases later. This would indicate that neither project is efficient at catching faults early in peer reviews but instead are catching them much later in test.

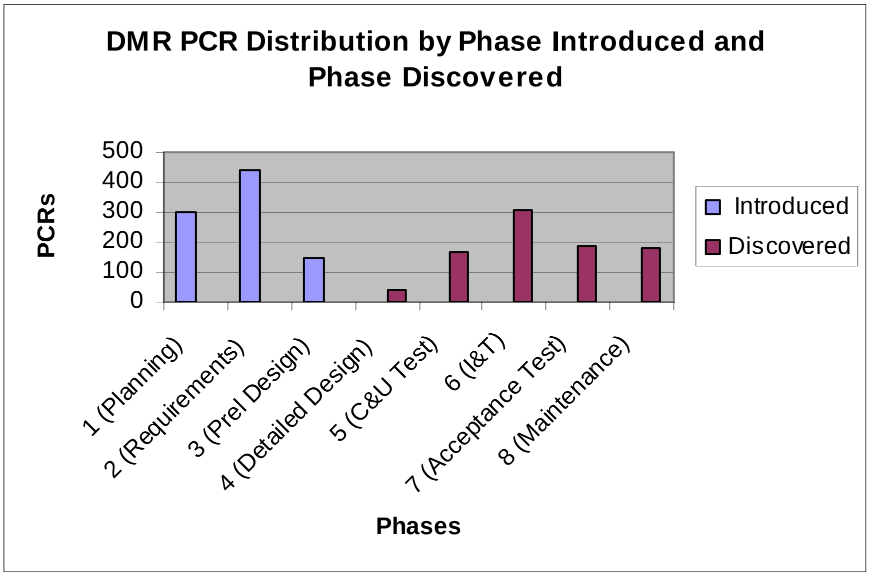

Figure 3: DMR PCR Phase Distribution

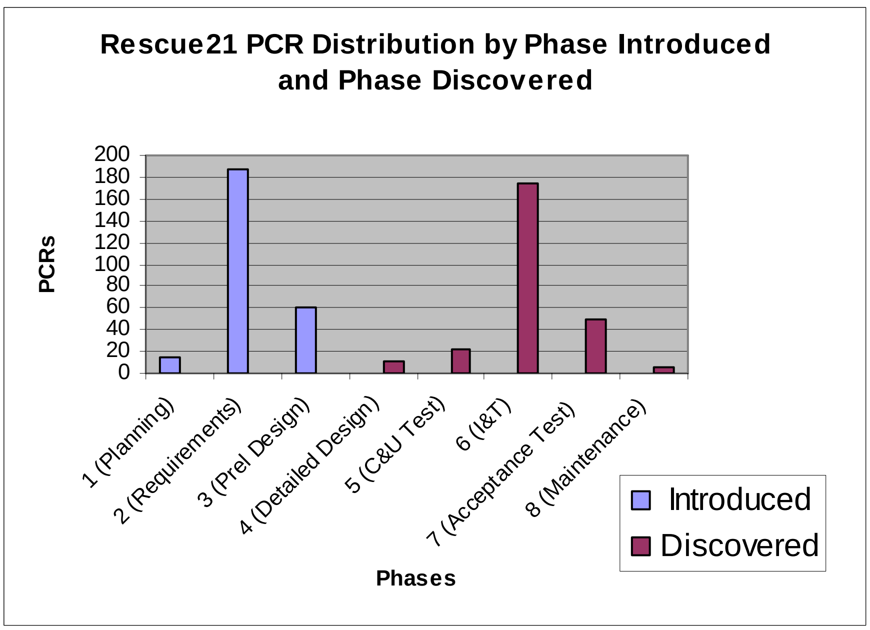

Figure 4: Rescue 21 PCR Phase Distribution

Results: Phase Containment Metric

Phase Containment was also computed using the PCR data. The aim of this metric is to show the phase differential, or number of phases between phase introduced and phase discovered. This number must be between one and seven, as each defect examined in this project was found at least one phase after it was introduced in order to be considered CRUD, and there are a total of eight phases in the engineering lifecycle. For this project, the phase differential was calculated for each PCR for both DMR and Rescue 21, and then it was plotted graphically for all PCRs. Phase Containment for DMR can be seen in Figure 5, and for Rescue 21 in Figure 6.

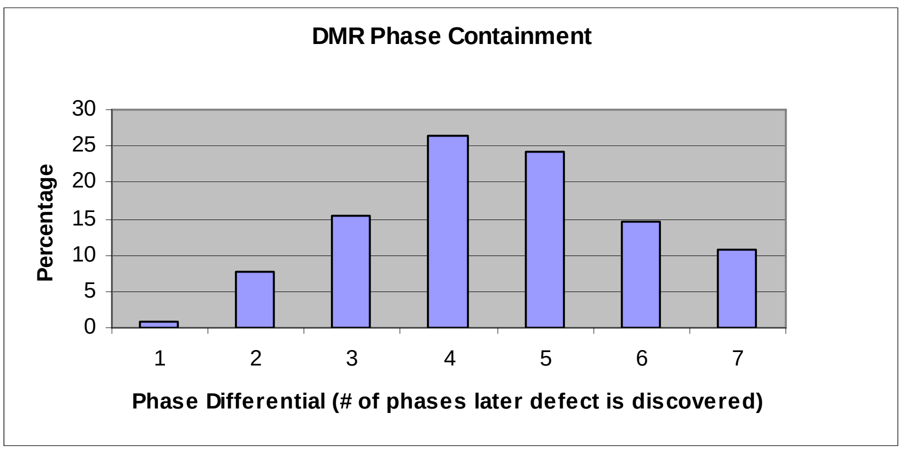

Figure 5: DMR Phase Containment

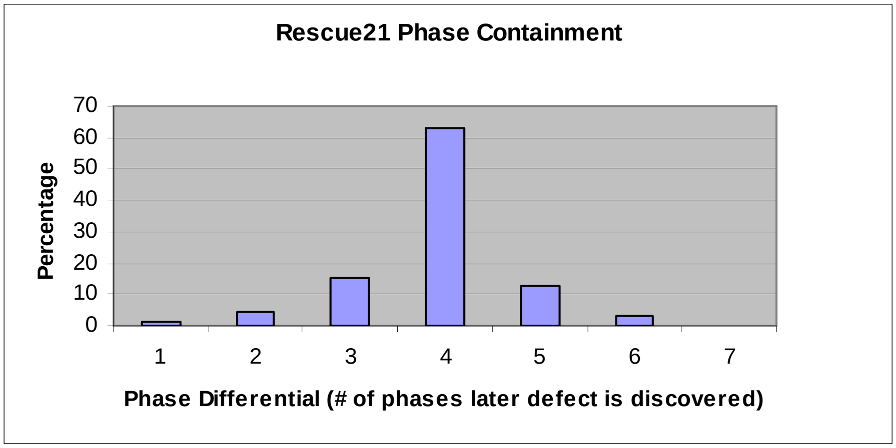

Figure 6: Rescue 21 Phase Containment

Both DMR and Rescue 21 have a high percentage of PCRs discovered at a differential of 4 or greater. When the bars in the 4+ range for each project are summed, the percentages are approx. 76% and 68% respectively. An important thing to note here is that neither labor nor defect data is collected by increment. What this means is that these numbers represent the *least* number of phases between introduction and discovery of a defect. For example, a defect may have been introduced in the system analysis phase (phase 1) of Increment 1 and discovered in the code and unit test phase (phase 5) of Increment 2, and it would only show up here as a three phase differential instead of a nine phase differential.

As stated before, the Phase Distribution and Phase Containment metrics reveal information about the efficiency of the project. In looking at these metrics for DMR and Rescue 21, it becomes clear that a large number of defects are found *out of phase*, most often four phases or later. This would indicate a lack of efficiency at the processes that uncover defects in-phase, such as peer reviews. A key issue may be system engineering participation in software reviews later in the engineering cycle. System engineering may be writing correct requirements and doing good preliminary design, but lack of participation in peer reviews from phase 4 of the engineering lifecycle on would lead to incorrect software design and implementation. Therefore, it is important for the proposed system engineering CRUD metric to examine the kinds of activities that system engineering is working on, not just the hours spent working.

Results: Proposed System Engineering CRUD Metric

Engineering hours in Gtime were examined for the proposed CRUD metric, approximately 60,000 records for both projects. First the hours were sorted by pay period. Then, the burden code was examined in order to discover which of these hours were spent on system engineering. For the DMR project, the second character of the burden code indicates the engineering discipline, and it is a number based on the engineering lifecycle (1-8). A number between one and three would hence indicate that the hours were spent on system engineering. For the Rescue 21 project, the third character of the burden code indicates the engineering discipline, and it is a letter based on the type of engineering that would be done. The letter “Y” indicates system engineering, the letter “S” indicates software engineering, and so on. The system engineering hours were separated out, as they were the only ones considered for this project.

Based on the results of the Phase Distribution and Phase Containment metrics for both projects, the system engineering hours then had to be sorted on what activity, or subprocess, was being worked on. Time spent by system engineers supporting non-systems work such as peer reviews is denoted in Gtime as evaluation, and is designated by the letter “V” in the burden code for both the DMR and Rescue 21 projects. Because the burden area field in a Gtime record indicates only this subprocess, the system engineering hours were then sorted by burden area. Hours spent evaluating were summed and represented as a percentage of total system engineering hours spent, per pay period.

The final step before graphing these hours and comparing them to CRUD was to shift the CRUD totals to the left. In Clearquest, only the date a defect is discovered is captured, but what is needed is the date a defect is introduced, as the time system engineering spends evaluating at the time a defect is introduced is what is important. In this case, the phase containment data for each project, along with the average time per increment for each project, provides a means to estimate the amount the data should be shifted. According to Program Management for DMR, the average increment time is nine months. This would mean that it takes approximately one month to complete each engineering lifecycle phase. Upon looking at Figure 6, it can be seen that most CRUD is discovered 4-5 phases after introduction. Therefore, the DMR CRUD data was shifted to the left by five months. The DMR system engineering CRUD over time can be found in Figure 7, and the system engineering hours spent evaluating over time can be found in Figure 8.

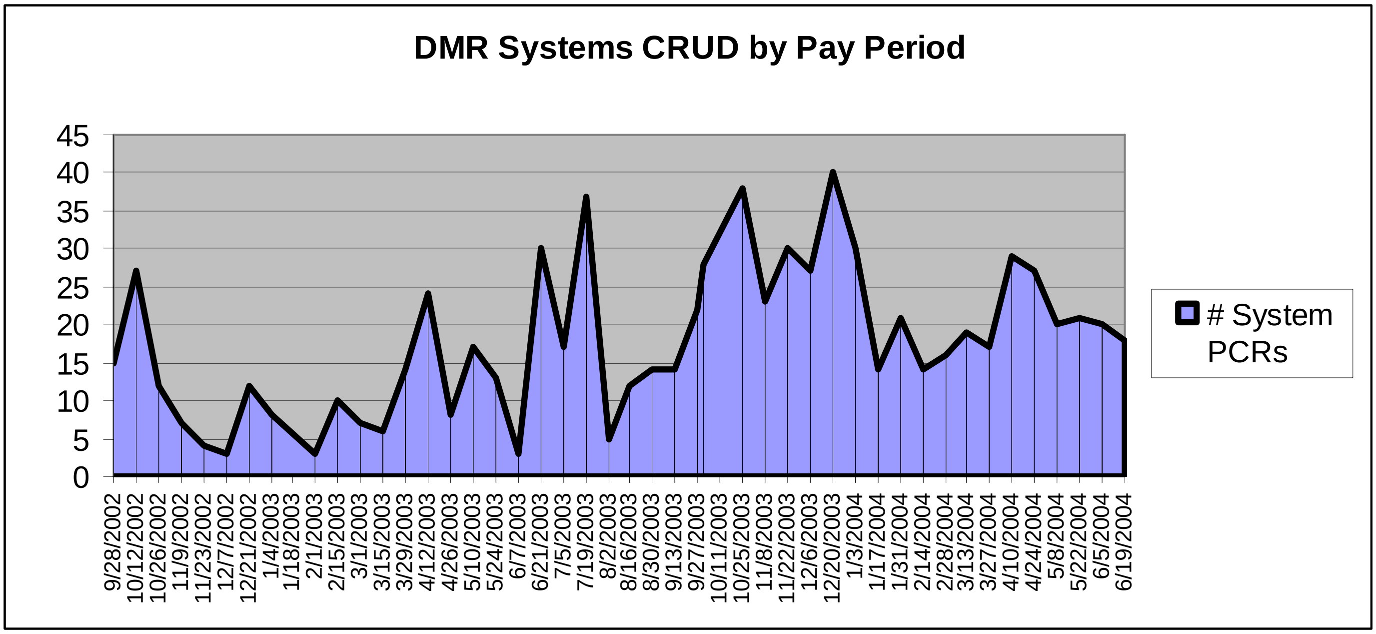

Figure 7: DMR Systems CRUD by Pay Period

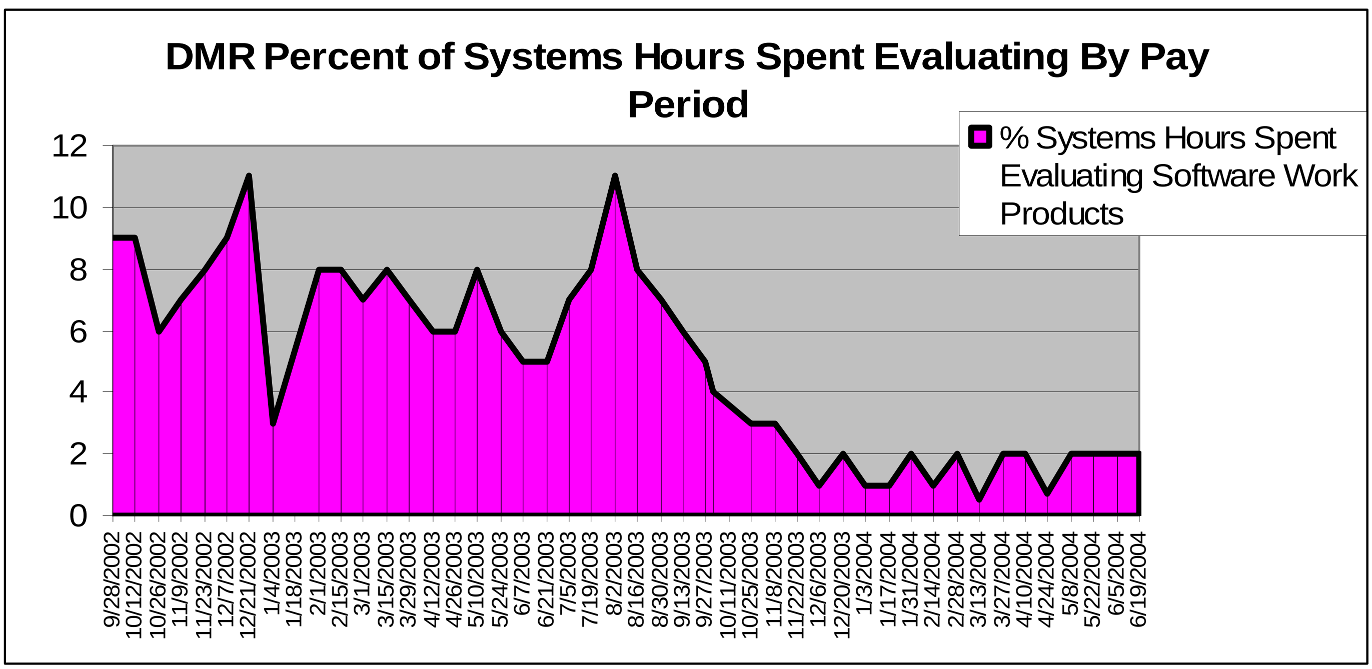

Figure 8: DMR System Hours Spent Evaluating by Pay Period as a Percentage of Total System Engineering Hours

In comparing DMR CRUD to DMR system engineering hours spent evaluating in Figures 7 and 8, a general trend can be seen. When evaluation hours are the lowest on average, from about 10/11/03 onward, CRUD increases on average. In addition, hours spent evaluating rarely hits 10%. This number may make sense in early system engineering phases of the lifecycle, when time is spent mainly generating system work products, but this number should be much higher in later phases.

For the Rescue 21 project, the phase containment data, along with the average time per increment, also was used to estimate the amount the CRUD data should be shifted to the left. According to Program Management for Rescue 21, the average increment time is twelve months. This would mean that it takes approximately one and a half months to complete each engineering lifecycle phase. Upon looking at Figure 6, it can be seen that most CRUD is discovered 4 phases after introduction. Therefore, the Rescue 21 CRUD data was shifted to the left by six months. The Rescue 21 system engineering CRUD over time can be found in Figure 9, and the system engineering hours spent evaluating over time can be found in Figure 10.

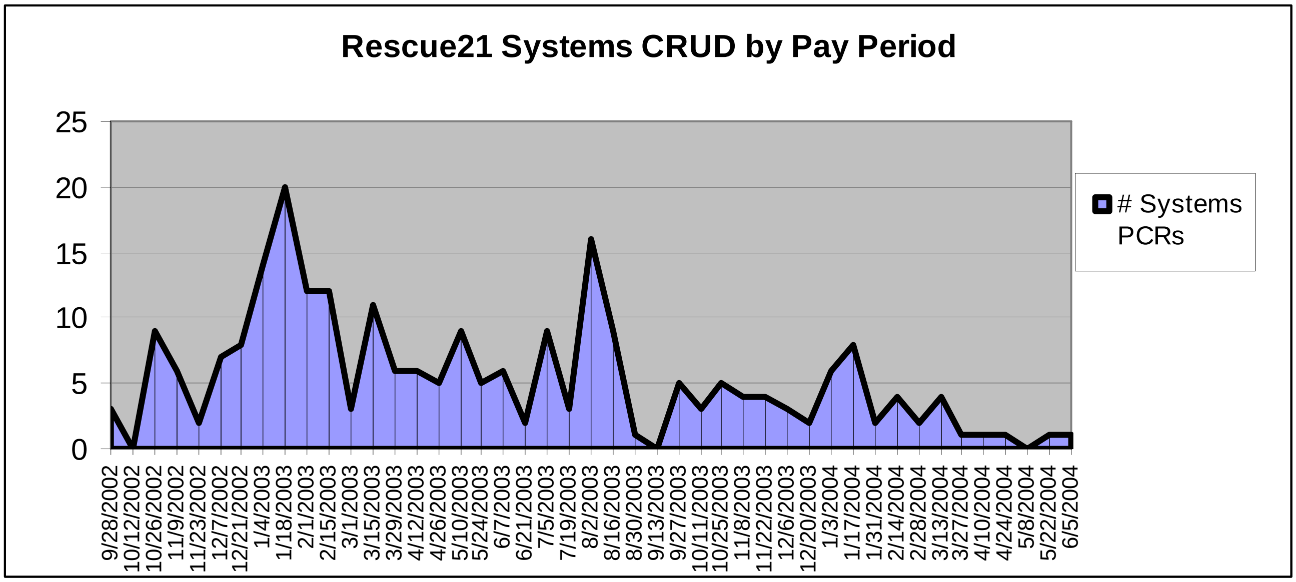

Figure 9: Rescue 21 Systems CRUD by Pay Period

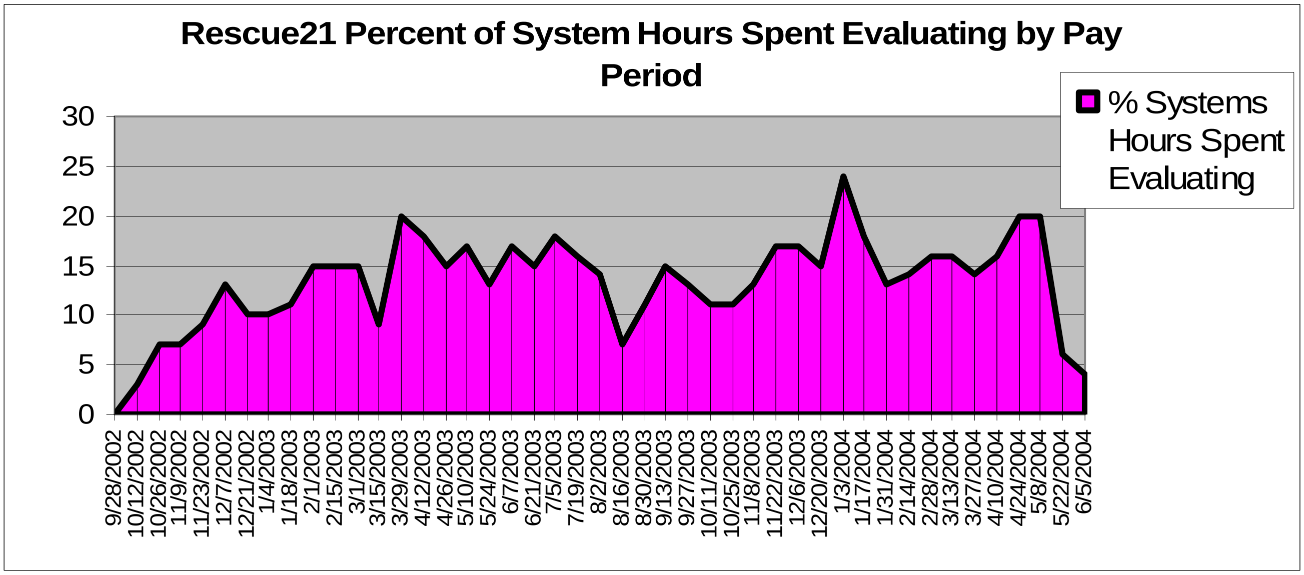

Figure 10: Rescue 21 System Hours Spent Evaluating by Pay Period as a Percentage of Total System Engineering Hours

In comparing Rescue 21 CRUD to Rescue 21 system engineering hours spent evaluating in Figures 9 and 10, the first thing to note is that, on average, system engineering hours spent evaluating is higher than in the DMR project. This could be because of the large amount of system engineering work on the project. However, hours spent evaluating rarely hits 20%. This number is still low for later phases of the engineering lifecycle, as system engineering leadership at GDC4S estimates that it should be as high as 50% in these phases. There is no general trend as there is for the DMR project in these charts; however, several peaks in the number of CRUD do correspond with dips in time spent evaluating.

Conclusions

The first conclusion to be drawn from this data analysis of two projects at General Dynamics C4 Systems is that there are too few useful system engineering metrics, particularly in the areas of efficiency and quality. Upon exploring two common software engineering efficiency metrics in the system engineering realm, data indicates that defects are being found many phases later than they should be, which increases cost. The likely cause of this lack of defect phase containment is that system engineering is not spending enough time supporting reviews in later phases in the lifecycle. This observation is supported by data for both projects showing a low percentage of system engineering hours spent evaluating over time. Trends in the data may become more evident if GDC4S were to collect data for both labor hours and product defects by increment. A recommendation would be to assign an increment number to each PCR, and to have a separate labor charge number for each increment in Gtime.

REFERENCES

1. International Council on System Engineering (INCOSE) System Engineering Center of Excellence (SECOE), http://www.secoe.org, 2005.
2. Software Engineering Institute (SEI) Capability Maturity Model® Integration (CMMI), http://www.sei.cmu.edu/cmmi/cmmi.html, 2005.
3. Rational Software, http://www.rational.com, 2005.
4. Chris Campbell, personal reference, 2005.

APPENDIX A

ACKRONYMS

ATR	Acceptance Test Rate
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CND	Communication Networks Division
COTS	Commercial Off The Shelf
CRUD	Customer Reported Unique Defects
DMR	Digital Modular Radio
GDC4S	General Dynamics C4 Systems
ID	Identification number
INCOSE	International Council on System Engineering
ISO	International Organization for Standardization
OTD	On-Time Delivery Performance
PCR	Process Change Request
SDR	Software Defined Radio
SECOE	System Engineering Center of Excellence
SEI	Software Engineering Institute
SNSD	Space and National Systems Division
WBS	Work Breakdown Structure

APPENDIX B

TYPICAL LIST OF SOFTWARE METRICS COLLECTED ON A PROJECT AT GDC4S

Metric	Category	Frequency	Trip Points
SEI Capability Maturity Model (CMMI) level	Process Maturity	Collected quarterly but reviewed monthly at Project reviews	< 5
SEI CMMI action plan	Process Maturity	Monthly Project Reviews	Action Plan must show corrective action vs scheduled action
Burden Codes	Effort Distribution	Collected and analyzed weekly by project	Rework > 15% Eval < 5% or > 20%
Latest Revised Estimate of SLOC over size	Product Size	Monthly Project Reviews	> 1.5 * original estimate
LRE SLOC/ BAC vs LRE SLOC/EAC or AIM Productivity Metrics	Needed Productivity	Monthly Project Reviews	Justify improved productivity needed
Staffing level planned vs actuals	Human Resources	Collected weekly and reviewed at Monthly Project Reviews	within 10% of plan
Cost and Schedule Variance	Earned Value	Reviewed at Monthly Project Reviews	CPI or SPI at 0.9
Percent Complete/ Level 5 Tool (units completed vs. planned) or equivalent	Work Performance	collect weekly analyze at Monthly Project reviews	within 15% of plan
Requirements Churn (req. changes / req.) using Level 5 tool or equivalent	Requirements Quality	Review at Monthly Project Review	exceeds project goal
SPCR Distribution by Root Cause using Level 5 tool	SPCR related	Review at periodic Metrics review with Quality Assurance (QA)	analyze top 3
Root Cause Effort Distribution using Level 5 tool	SPCR related	Review at periodic Metrics review with QA	analyze top 3
Phase Introduced and Discovered using Level 5 tool	SPCR related	Review at periodic Metrics review with QA	> 15% difference
Phase Containment Effectiveness using Level 5 tool	SPCR related	Review at periodic Metrics Review with QA	< 85%
CRUD	Product Quality	Review at Monthly Project Reviews	<= 1 defect per 5 KDSI
Reliability Metrics	Product Quality	Review at Monthly Project Reviews once Software Integration	<= 1 defect per 5KDSI, 100 hrs

Metric	Category	Frequency	Trip Points
		has started	MTBF for 50KSLOC goal,
Total and Closed Problems over Time	SPCR related	Review at Monthly Project Reviews	Evaluate rate of open problems
Peer Review SPC Charts	SPC	Review at periodic Metrics Review with QA, Stability and Capability issues/ corrective action to be presented at project reviews.	+/- 3 Sigma from mean 8 consecutive data points above or below the mean
TPM- RAM, ROM, Throughput	CpK	Review estimate vs requirement at Monthly Project Review	>= 1.5

APPENDIX C

TYPICAL LIST OF SYSTEM METRICS COLLECTED ON A PROJECT AT GDC4S

Metric	Goal Category	Update/Review Frequency	Project Goal	Trip Point
Capability Maturity Model (CMMI) level	Process Performance (Process Maturity)	Self Assessments updated quarterly & Reviewed monthly at Project reviews	All PAs at Level 5	<5
CMMI action plan	Process Performance (Process Maturity)	Action Plan is updated quarterly Reviewed monthly at Project reviews	Action Plan must show corrective action vs scheduled action	NA
Burden Codes	Process Performance (Effort Distribution)	Collected and analyzed monthly by project Org Level Reviewed qtrly by Chief Systems Engineer	Rework < 15% 5% ≤ Eval ≤ 20%	Rework > 15% Eval < 5% or > 20%
Cost and Schedule Performance Index	Process Performance (Earned Value)	Reviewed at Monthly Project Reviews	CPI or SPI > 0.9	CPI or SPI < 0.9
System Cumulative Requirements Volatility* using Level 5 tool	Product Quality (Requirements Quality)	Reviewed at Monthly Project Review	System Cumulative Requirements Volatility: ≤ 10%	System Cumulative Requirements volatility: > 10%
Customer Requirements Volatility	Product Quality (Requirements Quality)	Reviewed at Monthly Project Review	N/A	N/A
System PCRs	Product Quality (PCR related)	Reviewed quarterly as a part of the organizational level DPWG activities with CSE.	Analyze top 2 root cause codes	
Total and Closed Problems over Time	PCR related	Review at Monthly Project Reviews	Evaluate rate of open problems	

Metric	Goal Category	Update/Review Frequency	Project Goal	Trip Point
Staffing Levels Planned versus Actuals	Resources	Review at Monthly Project Reviews	> 10% off plan	Within 10% of Plan
Peer Review SPC Charts	SPC	Review at monthly Metrics Review with QA, Stability and Capability issues/ corrective action to be presented at monthly project reviews.	Stability Goal - +/- 3 Sigma from mean 8 consecutive data points above or below the mean	
Technical Performance Measures (TPMs)	Product Quality (CpK)	Review estimate vs requirement at Monthly Project Review	CpK >= 1.5	CpK < 1.5
Test Development Progress (Tests planned, generated, re-worked)		Review at Monthly Project Reviews		
Test Execution Progress (Tests planned, executed, passed)		Review at Monthly Project Reviews		

APPENDIX D

ENGINEERING TYPE DESCRIPTIONS

engineering type	description
system engineering	encompasses phase 1 through phase 3 of the engineering lifecycle, including planning, system level requirements analysis and description, and system level design
software engineering	encompasses phase 4 through phase 5 of the engineering lifecycle, including software level requirements analysis and description, and software level design and implementation
hardware engineering	considered as a parallel activity in the engineering lifecycle on primarily software/systems projects such as DMR and Rescue 21, includes all activities relating to the hardware platform of the system
other engineering	a labor hour category specific to the Rescue 21 project that is used as an administrative way to track engineering hours spent that may be charged back to a subcontractor or customer, not an actual type of engineering

APPENDIX E

DMR CHANGE REQUEST RAW DATA (SAMPLE)

Date_Opening	Headline	Phase_Introduced	Phase_Discovered
12/7/2003	Transmitter Sidetone Level	2 Requirements	7 Acceptance Test
12/7/2003	HMI pwr range incorrect for AM wfm	2 Requirements	7 Acceptance Test
12/8/2003	VxWorks "POSIX" Semaphores Only Count to 255	1 Planning	5 Code & Unit Test
12/8/2003	Sincgars Min/Max Frequency	2 Requirements	6 Integration & Test
12/9/2003	Remove HQ FH FIll Alarm reqt	2 Requirements	4 Detail Design
12/9/2003	Need Frequency Offset in Sincgars SC PT Voice Rx	3 Preliminary Design	6 Integration & Test
12/9/2003	Add mode for Sincgars SA	2 Requirements	6 Integration & Test
12/10/2003	182A Incorrect Test Level and Method APPSYS	2 Requirements	6 Integration & Test
12/10/2003	Link 4a: Update Appsys 3629 to convert SNR to BER	2 Requirements	6 Integration & Test
12/10/2003	SBI - 1200/2400 DESBPSK BER Degraded (SPAC01000148837)	3 Preliminary Design	7 Acceptance Test
12/10/2003	Update AM modulation level tests	3 Preliminary Design	6 Integration & Test
12/10/2003	Remove AM and FM Rx Selectivity tests	3 Preliminary Design	6 Integration & Test
12/10/2003	Fix WF test plan errors	3 Preliminary Design	6 Integration & Test
12/10/2003	Need to remove all KG84 FDEAR code from SATCOM	2 Requirements	6 Integration & Test
12/10/2003	Change APPSYS3631 to analysis	2 Requirements	7 Acceptance Test
12/11/2003	HMI need updated to COMSEC Setting screen	2 Requirements	6 Integration & Test
12/11/2003	Minor test plan updates required	3 Preliminary Design	6 Integration & Test
12/11/2003	Make 6.3 IRR test plan improvements	3 Preliminary Design	6 Integration & Test
12/11/2003	SBI -PA uses Rx frequency instead of Tx frequency	2 Requirements	6 Integration & Test
12/15/2003	Modify RF Black Hole Parameters for SC PT Voice	3 Preliminary Design	6 Integration & Test
12/15/2003	Change PA HW version check in Install Shield	2 Requirements	6 Integration & Test
12/16/2003	Include SDC 5.1.2 in Install Shield	2 Requirements	6 Integration & Test
12/16/2003	Must know if COMSEC or/and TRANSEC Key loaded	2 Requirements	5 Code & Unit Test
12/16/2003	Add compile flag for passthru or add/strip SA Info	2 Requirements	5 Code & Unit Test
12/16/2003	SBI - Frequency Scanning in the LOS waveform	2 Requirements	7 Acceptance Test
12/17/2003	SATCOM 182A Requirements at Incorrect Test Level	2 Requirements	6 Integration & Test
12/17/2003	SATCOM 182 Obsolete Requirements	2 Requirements	6 Integration & Test
12/17/2003	SATCOM 182A Requirement verific move to sub-build.	1 Planning	6 Integration & Test
12/17/2003	DCCC CCOW enum change in MIB Document	2 Requirements	6 Integration & Test
12/17/2003	Issue with Configure Baseband I/O port screen	1 Planning	6 Integration & Test
12/17/2003	Fill Device Type Compatibility and Testing	2 Requirements	7 Acceptance Test
12/17/2003	MIB variable name change in 6.3.0.1 INFRA DROP#2	2 Requirements	5 Code & Unit Test
12/18/2003	MIB security annotations update	2 Requirements	5 Code & Unit Test
12/18/2003	Red Phone CT/Indication during AutoClear Receive	2 Requirements	7 Acceptance Test
12/19/2003	SATCOM 182A Requirements at Incorrect T.L. Systems	2 Requirements	7 Acceptance Test
12/19/2003	SATCOM 182A Requirement Incorrectly Stated	2 Requirements	7 Acceptance Test
12/19/2003	remove sensitivity to the set of presets	3 Preliminary Design	7 Acceptance Test

APPENDIX F

RESCUE 21 CHANGE REQUEST RAW DATA (SAMPLE)

Discovered_date	Headline	Phase_introduced	Phase_discovered
6/11/2003	Configure Spectrum to poll R21 Applications	1- Planning	6- SW Integration & Test
6/11/2003	Modify Conferencing SPS - replay of historical data	2- Requirements Definition	6- SW Integration & Test
6/11/2003	Modify Asset Tracking SPS - replay of historical data	2- Requirements Definition	6- SW Integration & Test
6/11/2003	Modify Caller Position SPS - replay of historical data	2- Requirements Definition	6- SW Integration & Test
6/11/2003	Modify System Management SPS - replay of historical data	2- Requirements Definition	6- SW Integration & Test
6/11/2003	Modify Direction Finding SPS - DB access when opening DF Configuration panel	2- Requirements Definition	6- SW Integration & Test
6/16/2003	Modify Incident Tracking SPS - automatic closing of incident based on current time > end time	2- Requirements Definition	6- SW Integration & Test
6/18/2003	Association/Disassociation of DSC Location Reports Doesn't Handle MMSI ID Correctly	1- Planning	6- SW Integration & Test
6/18/2003	Modify Geo Display SPS - replay of historical data > 30 days	2- Requirements Definition	6- SW Integration & Test
6/19/2003	Asset Tracking 5.48 thru 5.50 - display of Lat/Long on Asset Tracking panels	2- Requirements Definition	6- SW Integration & Test
6/19/2003	Modify Direction Finding SPS - Restore Defaults operation should read DB-defined default values	2- Requirements Definition	6- SW Integration & Test
6/19/2003	SPS Need to be modified to remove failure scenarios that can not be done	2- Requirements Definition	6- SW Integration & Test
6/19/2003	SPS Needs to be updated for the following scenarios:	2- Requirements Definition	6- SW Integration & Test
6/20/2003	TC021.02.01-Install Software System Component	2- Requirements Definition	6- SW Integration & Test

APPENDIX G

DMR PROJECT HOURS RAW DATA (SAMPLE)

PyrlPerEndDt	LbrChrgNbr	RegHrQty	BurCd	Burden Area
9/21/2002	1631001000	2.5	Q0V	Q
9/21/2002	1631002200	3	S8	S
9/21/2002	1631004400	17	K86	K
9/21/2002	1631002200	25.5	S8	S
9/21/2002	1631004300	0.5	V6	V
9/21/2002	1631004400	1	S5	S
9/21/2002	1631004300	1	V5	V
9/21/2002	1631003200	1.5	G5	G
9/21/2002	1631003500	1.5	V5	V
9/21/2002	1631003400	1.5	V5	V
9/21/2002	1631004300	2	V5	V
9/21/2002	1631003650	3	V4	V
9/21/2002	1631003110	4	S5	S
9/21/2002	1631004400	4	V5	V
9/21/2002	1631004300	5	K5	K
9/21/2002	1631004300	6	V5	V
9/21/2002	1631004200	6.5	V5	V
9/21/2002	1631004400	7	G5	G
9/21/2002	1631002320	7	S5	S
9/21/2002	1631002200	8.5	G5	G
9/21/2002	1631003520	9	G4	G
9/21/2002	1631002360	10	S5	S
9/21/2002	1631003500	10.5	G5	G
9/21/2002	1631004300	11.5	G5	G
9/21/2002	1631004300	12	V4	V
9/21/2002	1631003650	13	G4	G
9/21/2002	1631002300	14	G4	G
9/21/2002	1631004300	15	G5	G
9/21/2002	1631003500	15	G5	G
9/21/2002	1631002320	17	G4	G
9/21/2002	1631001000	17	S5	S
9/21/2002	1631004300	18	G5	G
9/21/2002	1631004200	18	G5	G
9/21/2002	1631004400	18	V5	V
9/21/2002	1631004300	21	G5	G
9/21/2002	1631003300	21	G5	G
9/21/2002	1631004300	23.5	G4	G
9/21/2002	1631004500	23.5	K5	K
9/21/2002	1631002200	24	G4	G
9/21/2002	1631003650	26	G5	G

APPENDIX H

RESCUE21 CHANGE PROJECT HOURS RAW DATA (SAMPLE)

PyrlPerEndDt	LbrChrgNbr	RegHrQty	BurCd	Burden Area
9/28/2002	1627702100	3	GAY	G
9/28/2002	1627604720	4	GAY	G
9/28/2002	1627601101	12.1	GAY	G
9/28/2002	1627601101	16	GAY	G
9/28/2002	1627604720	26.5	GAY	G
9/28/2002	1627601121	2.9	GBY	G
9/28/2002	1627604740	6	GBY	G
9/28/2002	1627601121	9.5	GBY	G
9/28/2002	1627604740	9.9	GBY	G
9/28/2002	1627604720	12	GDY	G
9/28/2002	1627604720	16	GDY	G
9/28/2002	1627702400	0.5	GPY	G
9/28/2002	1627702400	0.8	GPY	G
9/28/2002	1627604710	1	GPY	G
9/28/2002	1627702100	1.6	GPY	G
9/28/2002	1627702100	2.8	GPY	G
9/28/2002	1627601101	6.4	GPY	G
9/28/2002	1627601101	8.5	GPY	G
9/28/2002	1627601101	11.4	GPY	G
9/28/2002	1627601101	15	GPY	G
9/28/2002	1627702400	0.7	SNY	S
9/28/2002	1627702100	2.5	SNY	S
9/28/2002	1627604750	6	SNY	S
9/28/2002	1627601103	9.5	SNY	S
9/28/2002	1627601101	10.3	SNY	S
9/28/2002	1627604710	12.5	SNY	S
9/28/2002	1627601101	13.5	SNY	S
9/28/2002	1627601101	25	SNY	S
9/28/2002	1627604710	23	SPY	S