

# **Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process**

R.R. (Ron) Willis  
R.M. (Bob) Rova  
M.D. (Mike) Scott  
M.I. (Martha) Johnson  
J.F. (John) Ryskowski  
J.A. (Jane) Moon  
K.C. (Ken) Shumate  
T.O. (Thomas) Winfield

Raytheon Systems Company

*May 1998*

TECHNICAL REPORT  
CMU/SEI-98-TR-006  
ESC-TR-98-006





CarnegieMellon  
**Software Engineering Institute**

---

Pittsburgh, PA 15213-3890

# **Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process**

CMU/SEI-98-TR-006  
ESC-TR-98-006

R.R. (Ron) Willis  
R.M. (Bob) Rova  
M.D. (Mike) Scott  
M.I. (Martha) Johnson  
J.F. (John) Ryskowski  
J.A. (Jane) Moon  
K.C. (Ken) Shumate  
T.O. (Thomas) Winfield

Raytheon Systems Company

*May 1998*

Unlimited distribution subject to the copyright.

This report was prepared for the

SEI Joint Program Office  
HQ ESC/DIB  
5 Eglin Street  
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

FOR RAYTHEON SYSTEMS COMPANY

(signature on file)

(signature on file)

Mario Moya, Maj, USAF  
SEI Joint Program Office

Terry Snyder, Senior Executive  
Raytheon Systems Company

This work is sponsored by the U.S. Department of Defense.

Copyright 1998 by Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Asset Source for Software Engineering Technology (ASSET): 1350 Earl L. Core Road; PO Box 3305; Morgantown, West Virginia 26505 / Phone: (304) 284-9000 or toll-free in the U.S. 1-800-547-8306 / FAX: (304) 284-9001 World Wide Web: <http://www.asset.com> / e-mail: [sei@asset.com](mailto:sei@asset.com)

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / Attn: BRR / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218 / Phone: (703) 767-8274 or toll-free in the U.S.: 1-800 225-3842.

---

# Table of Contents

<b>Abstract</b>	<b>ix</b>
Acknowledgment	ix
<b>1. Executive Summary</b>	<b>1</b>
Improvement Paradigm	1
Proven Process	2
Call to Action	2
Company-Wide Adoption of the CSWP	3
Continuing Improvement Plans	4
Positive Impact to Hughes Business	4
Positive Influence	5
<b>2. The Organization That Won the Award Was Hughes Aircraft Company, Software Engineering</b>	<b>7</b>
<b>3. Hughes Has a Long-Term Commitment to Software Process Improvement</b>	<b>11</b>
<b>4. Our Common Software Process Has Been Growing for a Quarter of a Century</b>	<b>17</b>
CSWP Content and Expansion	20
CSWP Change Process and Getting Consensus	22
Evolving Processes at Levels 4 and 5	24
Why the CSWP Has Survived and Grown	26
Assessment Results	26
<b>5. Deployment of the Common Software Process Is Happening Throughout Hughes Aircraft Company</b>	<b>29</b>
Process Deployment Team	29
The Twelve-Step Program	31
Software Process Assessment Approach	35
<b>6. We Have Many Associated Assets Supporting the Common Software Process</b>	<b>37</b>

<b>7. The Cornerstone of Our Process Is the Program Review</b>	<b>39</b>
<b>8. Performance Improved After Institutionalizing the Common Software Process</b>	<b>49</b>
The CSWP Helped Achieve CPI = 1.0 Within a Small Variation	49
Higher Maturity Yields Better CPI	52
CPI/SPI History for the Peace Shield Project	53
Review Efficiency Improved	53
Productivity Improvement of 987 Staff-Days	55
<b>9. Our Customers Are Pleased</b>	<b>57</b>
Improved Cost, Schedule, and Quality on Programs	57
Direct Customer Feedback	58
Other Aspects of Satisfying the Customer	59
<b>10. The Software Engineering Community Has Benefited From Our Investments in Software Process</b>	<b>61</b>
Presentations and Briefings	61
Books	71
Partnerships	72
Professional Associations	72
Contributions	74
Benchmarks	75
Recognitions	75
Innovations	76
Additional Results of Our Process Improvements	77
<b>11. Abbreviations</b>	<b>79</b>
<b>Bibliography</b>	<b>85</b>

---

# List of Figures

Figure 1 The Hughes Aircraft Company Software Organization. The Systems and Software Engineering Council oversees the process deployment activities 8

Figure 2: Geographical Distribution of Trained Software Process Assessors. Trained assessors are located throughout the organization 10

Figure 3: Origins of the CSWP. The CSWP evolved from documented processes that were developed to meet the needs of the Combat Grande program 18

Figure 4: Number of Users of Hughes CSWP. The number of users of the common process has grown over the past two decades, with significant increases in the past few years. 19

Figure 5 CSWP Practices and Example Procedures. The practices constitute high-level directives to implement Hughes common processes, while the procedures provide process and product implementation guidance. 21

Figure 6 Change Process for the CSWP. The process provides for improvements proven on projects or piloted in organizations to be incorporated into the CSWP. 22

Figure 7 Change Submittal and Review Process. The CSWP librarian maintains configuration control over change records and masters. 24

Figure 8 Process of Incorporating Level 4 and 5 Processes into the CSWP. Level 4 and 5 processes piloted and evaluated within designated Hughes organizations are incorporated into the CSWP after completion of the piloting effort. 25

Figure 9 Ongoing Improvement in Assessment Results. Over time, assessed organizations continue to improve their processes, while new organizations undergo baseline assessments to enable them to plan their subsequent process improvement programs.	27
Figure 10 Teams Involved in Process Definition and Deployment. The PDT provides a Hughes-wide forum for coordination of process assets, deployment, and assessments.	30
Figure 11: The Twelve-Step Program for Software Process Improvement. These steps provide a framework to guide progress in organizations new to process improvement	31
Figure 12 Status of Organization Process Improvement Activities. Status is tracked by the SEPG and reviewed with the sponsor.	34
Figure 13 Status of Project Process Improvement Activities. Status is tracked and reported to the SEPG for each project, then collectively reported to the sponsor for management review.	35
Figure 14 Training Assets Support the Common Software Process	38
Figure 15 Program Reports provide a variety of individuals with data to support a variety of needs, both within the program and in the organization.	42
Figure 16 The Team-of-Four concept provides the support mechanism needed to help ensure program success.	43
Figure 17 The MAT, ToF, and program review provide the SPM with several opportunities for visibility and support each month.	44
Figure 18 Organizational Schedule Performance and Cost Performance Indices indicate how the organization is performing against schedule and financial plans. This	



organization has been stabilizing and improving over the past two years.	46
Figure 19 Getting the status of the content of each report and metric helps to focus management attention on the key issues of each program.	46
Figure 20 Tracking how well the organization is doing in the collection and reporting of data is a reflection on the deployment of the process. Note the significant increase during the past eight months. This was a direct result of the establishment of the MAT.	47
Figure 21 Productivity Measurement provides the bottom line to our improvement activities.	47
Figure 22 CPI and SPI were optimized by process improvements during the early 1980s and following SEI-assisted assessments in 1987 and 1990.	51
Figure 23 Correlation between CPI and Process Maturity. As process maturity improves, CPI values cluster around 1.0 with minimal variance.	52
Figure 24 CPI and SPI Performance on Peace Shield. Predictability and continuous optimization of the software process resulted in a \$50,000,000 performance bonus.	53
Figure 25 Analysis of Quality Indicator data resulted in requirements review efficiency improvements that reduced overall effort.	54
Figure 26 Five-Year Productivity Improvement Goal Achieved in Three Years. Process maturity increased by one level during the performance period.	56
Figure 27 20 Years of Pleased Customers - As Evidenced by Letters of Commendation - 1977 to 1997.	60



---

# List of Tables

Table 1: Hughes Organizations With Trained Assessors	9
Table 2: History of Software Process Improvement	11
Table 3: Corporate Software Initiatives Teams	14
Table 4: Program Reporting, proven to be one of the most important ingredients in maturity improvement.	41



---

# Abstract

This report describes the software improvement activities of Hughes Aircraft Company<sup>1</sup> over the last 25 years. The focus is on continuous improvement of the software development process and the deployment of that process from a single organization at Fullerton, California, to virtually all the 5000 software engineers of Hughes Aircraft. For this achievement, the widespread deployment of a continuously improving software process, Hughes Aircraft was awarded the 1997 IEEE Computer Society Software Process Achievement Award.

## Acknowledgment

The authors would like to acknowledge the contributions of the hundreds of engineers and managers who, over a quarter of a century, developed, improved, and deployed the software development processes used at Hughes. We would like particularly to acknowledge the leadership of Terry Snyder. He was there at the beginning of the process improvement journey, inspired and demanded improvement throughout the effort, and was the key figure in the deployment of the process across the many organizations of Hughes Aircraft.

---

<sup>1</sup> At the time of the submittal of this paper, the authors were with Hughes Aircraft Company, which has recently merged with Raytheon.



---

# 1. Executive Summary

This report describes Hughes Aircraft Company's achievements that led to the 1997 selection for the IEEE Computer Society Award for Software Process Achievement. The achievement is *widespread deployment of a continuously improving software process* including:

- creation of a Capability Maturity Model<sup>SM</sup> (CMM<sup>SM</sup>)- and ISO-compliant software process with proven effectiveness
- deployment of this process throughout all of Hughes Aircraft Company
- significant positive influence on Hughes, industry, government and academia

Our software process is proven by 20 years of use and continuous improvement to be of high quality and supportive of performing to commitments. In 1990, Hughes was the first organization to achieve an SEI-assisted assessment rating of level three.

But, as good as the process is, only when it is institutionalized in an organization can we claim achievement. We evolved the process into the Hughes Common Software Process (CSWP) consisting of 22 Practices as shown in Figure 5. The CSWP has been adopted by 4800 software engineers in the 13 divisions that produce 85% of Hughes Aircraft's software as shown in Figure 9. More than 3000 engineers are in divisions assessed at levels 2, 3, and 4.

Improvement plans are being implemented in all of the divisions. As new adopters mature, we are seeing the same process quality and organizational performance as was measured extensively during the formative stages in the 1970s and 1980s.

## Improvement Paradigm

Our improvement culture was established in the early 1970s. While it has been called different names—e.g., Total Quality Management (TQM), Kaizen (relentless improvement), Zero Defects—the underlying theme for all of these descriptions is: *continuous measurable improvement (cmi)*.

Never settle for “good enough.” Always meet or exceed your customer's expectations. Fix the process, don't blame the people. Can this new process help us? Will the Software Engineering Institute (SEI) assessment help us find a better way? This attitude sums up our improvement culture.

## Proven Process

Hughes Aircraft pioneered the definition and measurement of software process to arrive at today's CMM- and ISO-compliant, efficient software process:

- Work on process development started in the 1970s.
- Early versions evolved through the use of teams of experts and lessons learned. A process maturity of level 2 was achieved in 1987 by an SEI-assisted assessment.
- Measurement became the basis for proving the capability of the DoD-STD-2167A based Software Engineering Practices and Procedures (SEPP).
- In 1990, we were recognized nationally as the first organization assessed at level 3 by an SEI-assisted assessment.
- We implemented level 4 and 5 processes on selected projects.

In 1977, we began project reporting using standard reports. These reports have been improved and are part of the current CSWP. A summary of the reports is shown in Table 4. The real-time, closed loop project reporting process is what we believe to be the distinguishing characteristic between us and other organizations who have failed to achieve levels 3 and 4. It is described in Section 7.

During the 1980s, we focused on validating and diagnosing the performance of the process. Metrics and reports from the individual project reports were summed to the organization level and used for this purpose. As explained in Section 8, we used CPI (cost performance index) and SPI (software process improvement) data summarized at the organizational level to continuously hone the process to reliably produce CPI/SPI close to 1.0. Figure 23 also shows that CPI is highly correlated to maturity level, validating our use of CPI as an optimization measure.

After the 1987 SEI-assisted assessment, we chose software review efficiency (i.e., percent of defects found and fixed in the same phase in which they were created) as our efficiency metric, reasoning that as more and more defects are caught within the phase they were produced, fewer and less costly rework of latent defects would be required. The most dramatic improvement, detection of *requirements* defects in phase, went from 43% review efficiency to 84%. This data is illustrated in Figure 25. This improvement meant that our organization did *not* need to correct 1,249 latent requirements defects. As described in Section 8, we realized a savings of 987 staff-days from this improvement.

## Call to Action

In 1992 and 1993, defense budget cuts, downsizing, plant closures, and reorganization led to dissemination of our primary software center of excellence into the other Hughes organizations. To sustain the momentum we had achieved, we reacted vigorously:



- Hughes Aircraft's President issued a policy requiring process improvement across the company.
- A new position, corporate vice president for Systems and Software Engineering was established.
- The Systems and Software Engineering Council (SSEC) was formed to net together the systems engineering and software engineering process owners from all Hughes organizations.
- Advantage was gained by the distribution of key process-advocates from the software center of excellence into the other divisions.

In this way, Hughes has sustained maturity growth momentum and withstood restructuring in spite of the significant pressures reshaping the defense industry. This is illustrated by Figure 9 showing sustained and expanding assessments and process improvement from 1987 to today.

As acquisitions and mergers occur, software organizations merging into Hughes are of varying levels, reflecting the industry mix. This will continue to affect our overall process maturity mix as it has in the past. As organizations merge, we bring their leaders into the SSEC and share our experiences and process assets to facilitate growth in the maturity of all of our processes.

### **Company-Wide Adoption of the CSWP**

In 1990, software leaders within Hughes organized a multi-division team to develop a CSWP using the proven SEPPs as the basis.

There were two significant outcomes from this effort:

- a proven, high-quality, defined CSWP with buy-in across the company
- the momentum for propelling widespread deployment across all of Hughes

The CSWP is compliant with the SEI CMM and ISO 9000 as it pertains to software. The CSWP is available to all Hughes software engineers via the Hughes Intranet or via CD-ROM as part of the Hughes Engineering Process System—along with all of the process assets needed to effectively deploy a process to a large, geographically dispersed organization. Section 6 summarizes the process assets, including a listing of the 60 training courses for the CSWP.

To support effective deployment of the CSWP, organizations have assessment or improvement activities and Software Engineering Process Groups (SEPGs) employing more than 50 process engineers across the company.

Hughes has experienced these significant benefits in deployment of the CSWP because of the extensive process assets:

- **Faster adoption.** Our Tucson software organization went from level 1 to level 3 in less than three years by deploying the existing proven process rather than struggling to develop one while trying to deploy.
- **Ready to go training.** The baseline is already established.
- **Reuse of lessons learned.** We reuse with the help of expertise distributed into the other divisions.

The CSWP evolved from the baseline SEPPs via a change-control process. Change management for the CSWP has been in place for the last 5 years, including process change review and approval boards now chartered by the SSEC.

### **Continuing Improvement Plans**

The Hughes goal is: “All organizations at or above level 3 by the end of 1998 and at least one organization at level 5.” Our experience in maturity growth in Tucson demonstrates the feasibility of moving organizations to level 3 quickly and our leading organizations already have many of the higher maturity key process areas (KPAs) in place. The ability to implement effective measurement comes with higher levels of process maturity. Now, as we are getting more organizations to levels 3 and 4, we are putting more measures in place.

Figure 24 shows the CPI/SPI data from the recently completed Peace Shield project. This recent data validates that the CSWP retains its capability for effective cost performance, achieving a CPI/SPI of .99/.99 while delivering more than 1.5 million lines of code.

The Hughes Tucson organization recently reached level 3 and has compared their initial defect-detection rates with previous Hughes norms (See Section 7). A lower than normal code-review efficiency has been identified. This fact-based comparison has provided timely insight that is driving process improvement now—rather than much later after their own norms are developed.

The SSEC has recently instituted standard metrics to summarize cost, schedule, maturity, quality, and productivity performance indicators at the division and company levels. It is anticipated that these measures will form the basis for the next generation of process improvement in Hughes.

### **Positive Impact to Hughes Business**

There has been considerable positive impact on our business including:

- successful completion of 25 Software Capability Evaluations (SCEs) and multiple Software Development Capability Evaluations (SDCEs) and ISO certifications, three including TickIT

- new business opportunities because of our effective process – e.g., winning Peace Shield and WAAS
- benefit to more than 200 projects
- growing sales in software-intensive markets.

In Section 9, two impressive commendations from our customers for the performance of our process and team are shown.

- 1977 – Combat Grande – this program served as the basis for documenting the initial common software process (see Section 4)
- 1995 – Peace Shield – for delivery ahead of schedule of a large, complex (1.5M lines of code) Air Defense system

### **Positive Influence**

Engineering leadership in Hughes is nearing the completion of process definitions for other engineering disciplines including systems engineering. This work has been based on the lessons learned from the definition and deployment of the CSWP. In addition, we are defining and deploying high-quality processes for program management and integrated product development.

Use of the CSWP in other elements of Hughes Electronics is growing.

- Hughes Space and Communications Company has begun adoption.
- DELCO Electronics participates in the SSEC.

The Hughes achievement has had significant influence on others as well:

- on industry, as discussed in Section 10
- on the SEI (helped define levels 3, 4, and 5)
- on competitors (set “high water mark” to achieve)
- on associates (via Software Process Improvement Network [SPIN], papers, and so on)



---

## 2. The Organization That Won the Award Was Hughes Aircraft Company, Software Engineering

The organization that received this award is Hughes Aircraft Company. Hughes Aircraft Company is a leading supplier of defense electronics including electro-optical systems, communication systems, radar systems, and missiles. Software is integral to these products. In addition, Hughes is a leading supplier of software-intensive systems including air traffic control systems, air defense systems, satellite ground systems, image processing systems, training systems, and related services. Software projects range in size from small teams to very large, complex, multi-team, multi-company projects delivering millions of lines of code.

Figure 1 presents the Hughes software organization. The three segments of Hughes Aircraft Company (boxes with shadows) contain the software-oriented organizations shown, and have assigned process owners for systems engineering (SE) and software engineering (SW). These process owners are responsible for deploying the common processes within their segments. In addition, the process owners meet as members of the Systems and Software Engineering Council (SSEC) chaired by Hughes vice president of Systems and Software Engineering, Terry R. Snyder. The SSEC is a corporate-funded effort to continuously improve, monitor, and deploy the common systems and software engineering processes throughout Hughes Aircraft Company. The SSEC is managed by T.O. Winfield and run like a program—setting, implementing, and achieving yearly goals for improvement. Results are reported by Snyder to the president of Hughes Aircraft Company.

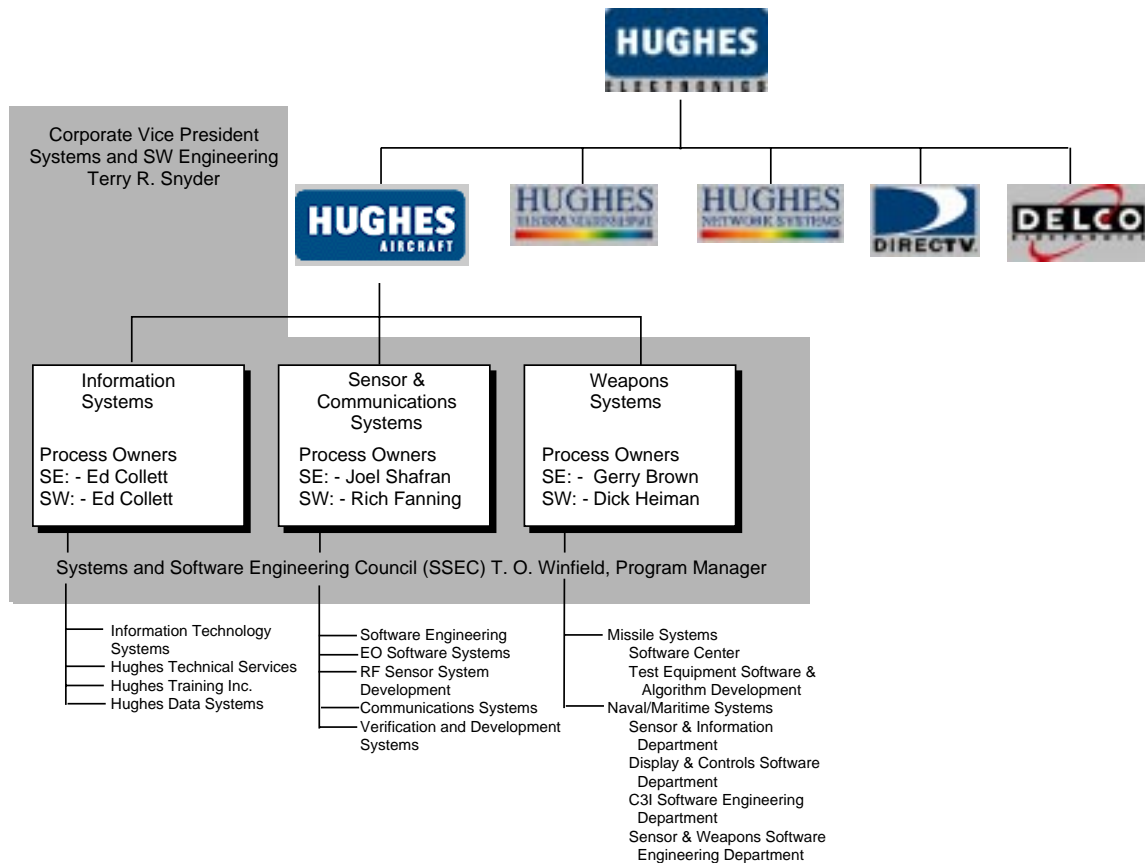


Figure 1 The Hughes Aircraft Company Software Organization. The Systems and Software Engineering Council oversees the process deployment activities

This figure is also a backdrop for understanding the senior management commitment, business management commitment, and project management commitment to software process improvement in our company.

- A corporate vice president of Systems and Software Engineering reports directly to the president of Hughes Aircraft Company.
- Over the last 10 years, \$23M in corporate funding was expended for process improvement.
- Process owners' networks work together on the common software process.
- 53 process engineers work in locally funded SEPGs networked company-wide.
- There is project-funded use and training of the CSWP.

Participation in the corporate software improvement effort to date has included 28 separate Hughes organizations. The organizations range from new acquisitions to long-term core elements of the company. A list of organizations that have sent individuals to Assessment Team Training is presented in Table 1. The geographical spread of these organizations is depicted in Figure 2.

<b>Hughes Organization</b>	<b>Location</b>
HITS-Civil Systems	Landover, Md.
HITS-Civil Systems	Washington, D.C.
HITS-Command & Control Systems	Fullerton, Calif.
HITS-Command & Control Systems	Reston, Va.
HITS-Defense Systems	Omaha, Neb.
HITS-Defense Systems	Reston, Va.
HITS-Space Systems	Denver, Colo.
Hughes Air Warfare Center	Indianapolis, Ind.
Hughes Consulting Group	Pontiac, Mich.
Hughes Danbury Optical Systems, Inc.	Danbury, Conn.
Hughes Defense Communications	Ft. Wayne, Ind.
Hughes Defense Communications	Torrance, Calif.
Hughes Sensors & Communications Systems - Optical	El Segundo, Calif.
Hughes Sensors & Communications Systems - Processor	El Segundo, Calif.
Hughes Sensors & Communications Systems - Radar	El Segundo, Calif.
Hughes Space and Communications Company	El Segundo, Calif.
Hughes STX	Lanham, Md.
Hughes STX	Sioux Falls, S.D.
Hughes Training Inc.	Arlington, Texas
Hughes Training Inc.	Binghamton, N.Y.
Hughes Training Inc.	Herndon, Va.
Hughes Training Inc.	Houston, Texas
Hughes Training Inc.	Orlando, Fla.
Hughes Training Inc.	Pontiac, Mich.
WSS HMSC Software Engineering Center	Tucson, Ariz.
WSS HMSC Test Equipment	Tucson, Ariz.
WSS Naval and Maritime Systems	Fullerton, Calif.
WSS Naval and Maritime Systems	San Diego, Calif.

*Table 1: Hughes Organizations With Trained Assessors*



*Figure 2: Geographical Distribution of Trained Software Process Assessors. Trained assessors are located throughout the organization*



### 3. Hughes Has a Long-Term Commitment to Software Process Improvement

Dates	Program	Focus	Results
1970 to 1989	Individual Business Unit Initiatives	<ul style="list-style-type: none"> <li>Improve software capability within individual business units.</li> </ul>	<ul style="list-style-type: none"> <li>Fullerton rated level 3</li> </ul>
1989 to 1992	Corporate Software Initiative (CSI)	<ul style="list-style-type: none"> <li>All software organizations must reach their appropriate SEI maturity level</li> </ul>	<ul style="list-style-type: none"> <li>several organizations moved from level 1 to 2</li> <li>30% reduction in the cost of tool acquisition</li> <li>joined Software Productivity Consortium</li> <li>assigned resident affiliates to SEI</li> </ul>
1992 to 1995	Software Technology Network (STN)	<ul style="list-style-type: none"> <li>Define a common approach to software development for major Hughes software organizations.</li> <li>Benchmark with other companies to identify leap-ahead technology.</li> </ul>	<ul style="list-style-type: none"> <li>more than 50% of engineers working in level 2 or 3 organizations</li> <li>adoption of the Common Software Practices</li> </ul>
1995 to Present	Systems and Software Engineering Council (SSEC)	<ul style="list-style-type: none"> <li>Deploy Common Software Practices to all Hughes software organizations.</li> <li>Define a common approach to systems engineering</li> </ul>	<ul style="list-style-type: none"> <li>85% of software engineers working in level 2 or 3 organizations</li> <li>better integration of systems and software engineering</li> <li>adoption of Systems Engineering CMM</li> </ul>

Table 2: History of Software Process Improvement

Hughes has had a commitment to software process improvement since the early 1970s. This very early commitment was demonstrated at the business unit level. However, in 1989, it was realized that a corporate commitment would benefit all software organizations in the company. This realization came about because it was felt that the lack of software engineering competency was a national problem and having all business units address the same problem on their own was not an optimum approach. Our customers were demanding better quality software at lower cost. While there had been significant progress in improving software capability in some of the software organizations (at the Fullerton Software Engineering Division in particular, as illustrated in “Software Process Improvement at Hughes Aircraft”, *IEEE Software*, July, 1991), Hughes corporate management viewed this as a challenge to the company as a whole, rather than local organizational concerns.

During the late 1980s, the company was reorganizing to meet the challenges of the Department of Defense (DoD) downsizing. Hughes management felt that our ability to develop software for the DoD and other government agencies was a core competency for the company. To that end, the Software Technology Advisory Council (STAC) was formed to define the approach to making Hughes competitive in the face of the new market realities. This body recommended the formation of the Corporate Software Initiatives (CSI) Program in late 1989.

The CSI Program had the charter to organize and lead a multi-organizational team whose overall goal was to gain greater competitive advantage through more effective application of software process, methods, and tools in development of software for our customer base.

This program was funded and managed from the corporate technology office. However, the staffing and planning was supplied by the operating units in the company. These were the units responsible for software development. This body was provided the necessary funding to initiate programs within Hughes to address the problems of building software for our diverse customer base. Various teams were formed to address the problem of defining and deploying mature software processes, and providing automation to that process. In addition, Hughes developed affiliations with other organizations responsible for improving the software process, such as the Software Productivity Consortium, MCC (Microelectronics & Computer Technology Corporation), UC (University of California) Irvine and University of Southern California (USC).

Given that it had been shown that there was a strong correlation between program success (cost, quality, and schedule) and the process maturity of an organization (see Section 8), it was determined that one of the primary thrust of the CSI program would be to raise the maturity level of the operating units. A stated goal of the CSI Program was to have all software organizations at or above level 2, with all software organizations with 100 or more software engineers at level 3 or above. The desire of Hughes management was to have a high level of maturity and competitiveness throughout the company, rather than have one or two high-level (showcase) organizations and many other low-maturity organizations.

Because of the autonomous nature of the operating units and the diverse customer base, we determined that each organization could develop their own approach to improving their processes. Rather than focus on a common software process for all of Hughes software, the focus would be on finding commonality in certain areas. Among the areas were cost and schedule estimating, risk management, requirements management and improved use of CASE (computer-aided software engineering) tools to support software development.

Teams, composed of members of each of the software organizations, were launched to address each of the focus areas.

<b>Team</b>	<b>Focus</b>	<b>Accomplishment</b>
Cost Estimating and Schedule Estimating	Define a standard approach to cost estimating based on the use of the SEER model and historical databases.	The proposal process in the software organizations were altered to accommodate this revised software estimating process. Strategic alliances were negotiated with cost model tool vendors to reduce the cost of tool acquisition.
Risk Management	Develop techniques to address software risk management.	The program management risk process was merged with the engineering risk management process to provide an integrated approach to addressing risk on programs. Comprehensive training was developed for risk management and customized tools developed to help quantify and manage risk.
Requirements Management	Select an approach to real-time structured analysis (RTSA).	Adopted the Hatley-Pirbhai RTSA methodology as the approach to requirements development. A set of commercial tools were selected to accommodate this methodology. A training program was established to train software and systems engineers in the use of the methodology and tool. More than 400 engineers were trained.
CASE Tools	Define a software	This team negotiated

Team	Focus	Accomplishment
	engineering environment to support the activities associated with software requirements analysis and design.	contracts with vendors that allowed Hughes to reduce the cost of acquiring tools by 30%. In addition, one member of the team worked with the SEI to define techniques for tool integration and define techniques to facilitate technology transfer for CASE tools.

Table 3: Corporate Software Initiatives Teams

In mid 1991, Hughes underwent another reorganization. This time the goal of the reorganization was to optimize the corporate staff. Many of the technology and other engineering functions were moved out of corporate into the operating elements of the company. In spite of this reorganization, the company realized the need to maintain a company-wide focus on software. This was because of the need to maintain software as a core competency and because of the significant progress made by the CSI Program. The company organized the Software Technology Network (STN) (along with other networks to address the technology needs of the company).

The objectives of the STN were to:

1. assist in the development of coordinated and integrated technology plans for software
2. develop coordinated and integrated plans for common design, development, and manufacturing methods and tools
3. develop alliances with consortia, universities, General Motors, et al
4. administer activities associated with general communication of technical information
5. assist competitive analyses and benchmarking efforts to evaluate our technological capabilities relative to market needs
6. participate in the establishment of relevant Centers of Excellence

While the focus of the STN remained process improvement, the same as the CSI program, other activities were initiated to find “leap ahead” approaches to software engineering. These approaches would be designed to find approaches to integrate promising (and less costly) commercial practices into the traditional DoD-related approaches used by Hughes in the past.

To accomplish this, the STN launched teams to study database engineering techniques and commercial practices. Hughes also began a comprehensive benchmarking program designed to share knowledge with other companies in order to find best practices. The results of these

benchmarking exercises led to the adoption of object-oriented techniques for software development that are being used in our software-intensive organizations.

During this time, to accelerate the process maturity effort, the operating units agreed to use one common approach to software development throughout Hughes because of the demonstrated maturity (or best practices) of the Fullerton organization. In addition, there were obvious areas of overlap in the organizational needs. For example, the approach to managing software projects could be common throughout the company. By adopting a common process, progress toward higher maturity could be made at a faster pace.

That approach was named the CSWP. The genesis of the CSWP was the software practices from the Fullerton, California, based Software Engineering Division. Plans were developed to convert all major Hughes organizations to the CSWP.

A team was organized to make the CSWP applicable to all major software organizations. This involved adopting language that was generic enough to apply to all organizations. Most organizations had different titles for members of software teams. Titles such as software team lead or software program manager were used interchangeably. In some cases, the products of the various phases of the software development process had different names and indeed, the phases themselves had different names. The team resolved these issues. This team also had the additional responsibility to act as the change control board (CCB) for the CSWP. All changes to the CSWP were addressed by this team.

In 1995, Hughes management organized all engineering disciplines (systems, software, electrical, mechanical, etc.) under the leadership of the Hughes Engineering Executive Council. As part of this effort, the STN was combined with the Systems Engineering Network (SEN) to become the Systems and Software Engineering Council (SSEC). A Hughes vice president was appointed to head the SSEC. The major objective of the SSEC was to continue (and accelerate) the deployment of the CSWP.

As part of the Hughes focus on process, a Process Management, Assessment and Standard Tools (PMAST) body was formed. This body was given the tasks of defining and implementing an approach for documenting and managing changes to the Hughes-wide processes. The body was also charged with determining how to measure the benefits of common processes. This included conducting assessments for all the engineering processes.

Process Owner Councils (POC) were established to be responsible for the definition, deployment, and maintenance of the various engineering processes. The SSEC was designated the POC for systems and software engineering processes.

The SSEC was given the additional responsibility to define and deploy a common systems engineering process and improve the process maturity of systems engineering. During this time, the SE CMM was developed and released by a team lead by the SEI. Hughes

participated on this team and was among the early users of the SE CMM to measure systems engineering maturity.

The SSEC, along with SEI, has trained several Hughes lead assessors. The SSEC uses these lead assessors to conduct CBA IPI (CMM-Based Appraisal for Internal Process Improvement) assessments of all major software organizations throughout Hughes. All software organizations are required to have periodic assessments. These assessments are used to develop action plans. The action plans are monitored by the SSEC.

In addition to the activities to improve the software process by increasing the software maturity of the software organizations, Hughes has been involved with many organizations whose objectives are to improve software process maturity.

Hughes was an early member of the Software Productivity Consortium (SPC). Hughes participated in the develop of the ADARTS methodology for the design and development of Ada software.

Hughes has had a long relationship with the SEI. There have been three resident affiliates. These affiliates have supported the development of CASE environment design and deployment, evolution of the Software and Systems Engineering CMMs, and techniques for technology change management.

In the Southern California area, Hughes is a charter member of the University of California at Irvine, Irvine Research Unit in Software (IRUS). This body manages one of the largest SPINs and addresses special research topics in software such as testing.

---

## 4. Our Common Software Process Has Been Growing for a Quarter of a Century

Starting in 1974, in the Computer Programming Laboratory of 120 software engineers, through 1987 in the Software Engineering Division of Ground Systems Group with more than 1000 users, until the present time with approximately 4800 software engineering users of the CSWP throughout Hughes Aircraft Company, there has been continuous growth in the use of the common process.

In the mid 1970s, the Combat Grande program served as the basis for documenting the initial common software processes. This national air defense system was produced by developers originating from many organizations and technical backgrounds who agreed to work to a common definition of technical processes to guide their development approach. The Combat Grande team worked closely with the customer on site in Europe, providing management reports for Hughes management and visibility for the customer into progress achieved. By 1980, the success of this early set of standards and guidelines implementing structured design methods, peer reviews, language standards, computer resource utilization, and configuration control for multiple baselines led to acceptance of the defined common process on projects throughout the newly formed Software Engineering Division (SED).

The processes and management methods defined for Combat Grande formed the earliest basis for today's common processes, as shown in Figure 3.

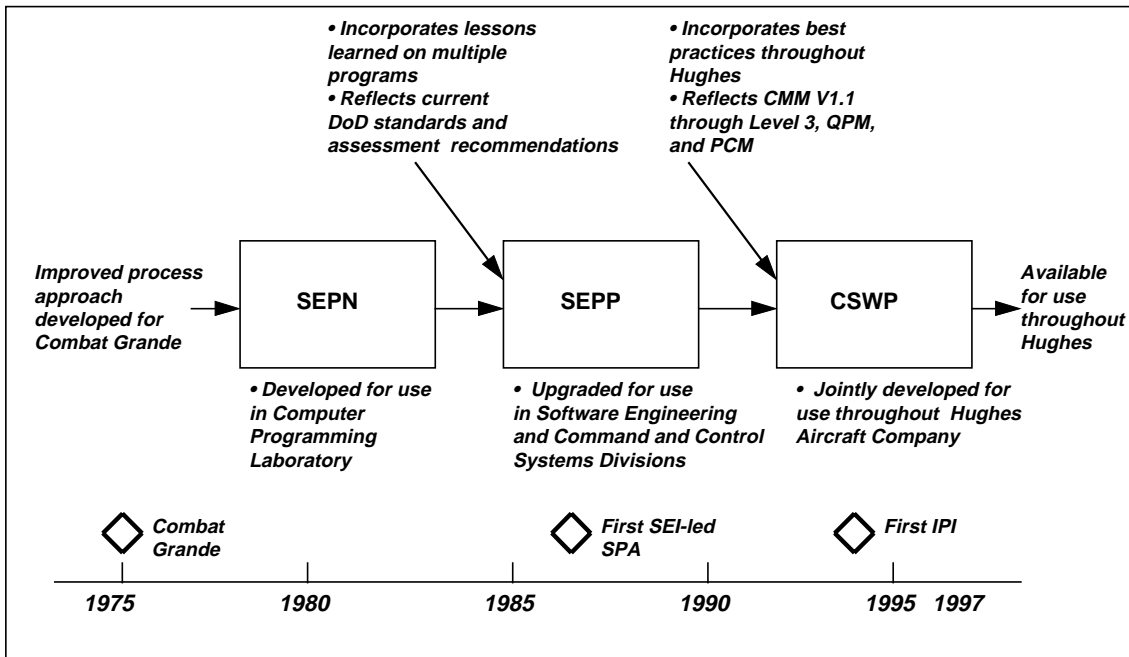


Figure 3: Origins of the CSWP. The CSWP evolved from documented processes that were developed to meet the needs of the Combat Grande program

With the evolution of military standards from MIL-STD-490 and -483 to DoD-STD-2167, internal Hughes SEPPs were developed to clarify the earlier processes and provide a more succinct set of standards in the mid 1980s. These documented individual software engineering phases and activities, clearly defining the design and development documents to be produced, and were closely aligned to contractual requirements. Technical and management metrics continued to evolve to a more comprehensive set of data represented in reports produced monthly and presented regularly to program managers and senior management. The success of the SEPPs was reflected in wide support from the customer community and in both improved productivity and quality in SED products. The number of users had grown to more than 1000 in the late 1980s.

Hughes SED first became affiliated with the SEI in early 1987 and agreed to participate in a pilot software process assessment (SPA). Several members of the SEI, including Watts Humphrey and Bill Curtis, trained the assessment team and served as team members. At the time of this assessment, the SEPPs were used on all SED projects and subsequently were introduced into the Command and Control Systems Division. The assessment introduced the SEI questionnaire (commonly referred to as TR-23) into the Hughes organization. Assessment findings led SED to expand the SEPPs to require establishment of an SEPG and define its role, and also led to many detailed improvements in the SEPP that helped the division to achieve its level 3 rating in 1990.

In the early 1990s, establishment of a Hughes-wide, corporate-sponsored software initiatives program led some other divisions to undergo software process assessments, achieving level 2



and 3 ratings, while SED, using its continuously maturing SEPP, moved on to a level 4 rating in an internal assessment.

The success of these assessments stimulated the collaboration of four major software engineering organizations in developing a core common software process, evolving the SEPP to a more broadly accepted CSWP. Based on the SEPPs, the CSWP incorporated “best practices” from participating organizations into a set of practices applicable to a broad range of organizations and projects. By 1994, the CSWP was in use by several expanding divisions, and the number of software engineering users of the CSWP had grown to more than 2100, as shown in Figure 4. The number of users has grown to more than 4800 as of this writing.

#### NUMBER OF USERS AT KNOWN MILESTONES

	Informal SEPNS	Formal SEPNS	SED	SEPPs	CSWP "Seed"	CSWP	CSWP	Today
	1974	1978	1983	1987	1992	1994	1996	1997
<b>Total Users</b>	120	450	900	1000	1300	2190	3575	4800
Computer Programming Laboratory (CPL)	120	450						
Software Engineering Division (SED)			900	600	400			
Command Control Systems Division(CCSD)				400	900	1300	1100	1300
SW Engineering Lab, Processor Division						135	130	130
SW Engineering Lab, Systems Division							375	375
Software Systems Center (EOS)						100	300	400
Hughes Missile Systems Company (HMSC)						200	300	350
Naval and Maritime Systems (NAMS)						80	120	175
HITS, Space Systems							680	700
HITS, Defense Systems							175	200
HITS, Civil Systems							245	300
HDOS							10	10
Hughes Training Inc. (HTI), Arlington							140	410
Hughes Training Inc. (HTI), Binghamton								200
Hughes Air Warfare Center (HAWC)								250

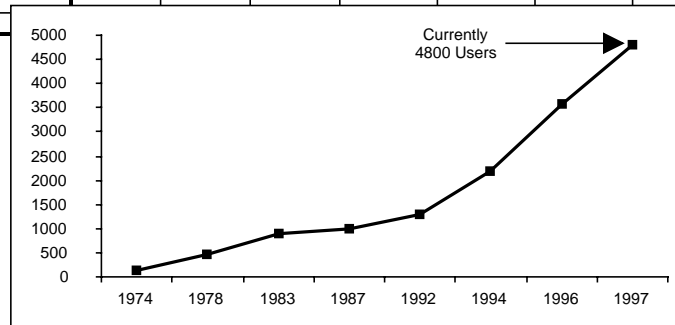


Figure 4: Number of Users of Hughes CSWP. The number of users of the common process has grown over the past two decades, with significant increases in the past few years.

Outstanding results in many SCEs and consistent success in achieving recognition for outstanding process achievements on programs such as Peace Shield and, more recently, Wide Area Augmentation System (WAAS) has provided added incentive for adoption of the CSWP in other Hughes organizations. Reduced risk and higher productivity, as well as improved management, provided the motivation that led software engineering organizations throughout the company to train both managers and development personnel on the use of the CSWP.

## **CSWP Content and Expansion**

As of mid 1994, the CSWP incorporated practices from the SEPPs and addressed requirements of the CMM Version 1.1. A core of 19 directives was supplemented by more than 40 detailed procedures, all fully compatible with the CMM.

Within the next two years, new features were incorporated into the CSWP, such as improvements accommodating specific ISO 9001 requirements, added life-cycle definitions, and the IEEE and other commercial standards replacing MIL-STD-498. Most recent additions address software reuse and COTS (commercial off-the-shelf software), with additional processes for maintenance and support of existing software. A list of specific practices defined in the CSWP is shown in Figure 5.

## COMMON SOFTWARE PROCESS (CSWP) PRACTICES

- 1 Software Directives System
- 2 Financial Management
- 3 Project Tracking and Reporting
- 4 Software Engineering Training
- 5 Software Process Engineering
- 6 Software Project Directive System
- 7 Software Configuration Management
- 8 Software Evaluation
- 9 Software Requirements
- 10 Preliminary Design
- 11 Detailed Design
- 12 Coding
- 13 Unit Testing
- 14 Integration and Testing
- 15 CSCI Testing
- 16 Other Software Documentation
- 17 Software Proposal and Cost Estimating
- 18 Software Subcontract Management
- 19 Software Quality Assurance
- 20 Software Coordination With Other Engineering Disciplines
- 21 Software Servicing and Maintenance
- 22 Software Reuse

## COMMON SOFTWARE PROCESS (CSWP)

### PRACTICE 3 – PROJECT TRACKING & REPORTING

#### PROCEDURES

- 3.2.1 Project Overview
- 3.2.2 Accomplishment Summary
- 3.2.3 Problem Summary
- 3.2.4 Project Schedule
- 3.2.5 Risk Status Report
- 3.2.6 Milestones Report
- 3.2.7 Rate Chart Report
- 3.2.8 Earned Value Report
- 3.2.9 Target System Resource Usage Report
- 3.2.10 Software Project Resource Forecast Report
- 3.2.11 Financial/Staffing Report
- 3.2.12 Quality Indicator Reports
- 3.2.13 Scope Change Report
- 3.2.14 Lessons Learned Report
- 3.2.15 Software Problems Status Report
- 3.2.16 Productivity Measurement Report
- 3.2.17 Software Size Trend Report
- 3.2.18 Defect Density Tracking Report
- 3.2.19 Software Requirements Volatility Report
- 3.2.20 Software Management Effectiveness

*Figure 5 CSWP Practices and Example Procedures. The practices constitute high-level directives to implement Hughes common processes, while the procedures provide process and product implementation guidance.*

The CSWP focuses on proactive processes: risk aversion, proactive management methods intended to prevent problems from occurring during a software development project, and prevention of defects in the software under development. The resulting process supports software development in an integrated product team environment or in a more conventional software organization and can be adapted to a selection of life cycles. This provides the flexibility needed by the wide variety of users and projects throughout Hughes Aircraft Company.

### CSWP Change Process and Getting Consensus

The CCB established for the CSWP, under the sponsorship of the Hughes Systems and Software Engineering Council, is the Software Engineering Process Team (SWPT). Membership in the SWPT is from each of the major segments of Hughes Aircraft Company and software quality assurance, creating a team of five persons. Change requests can be submitted by anyone within Hughes, reviewed within their own organizations, and then provided directly to the SWPT for review and approval. The change process implements the Hughes-wide cmi concept, as shown in Figure 6.

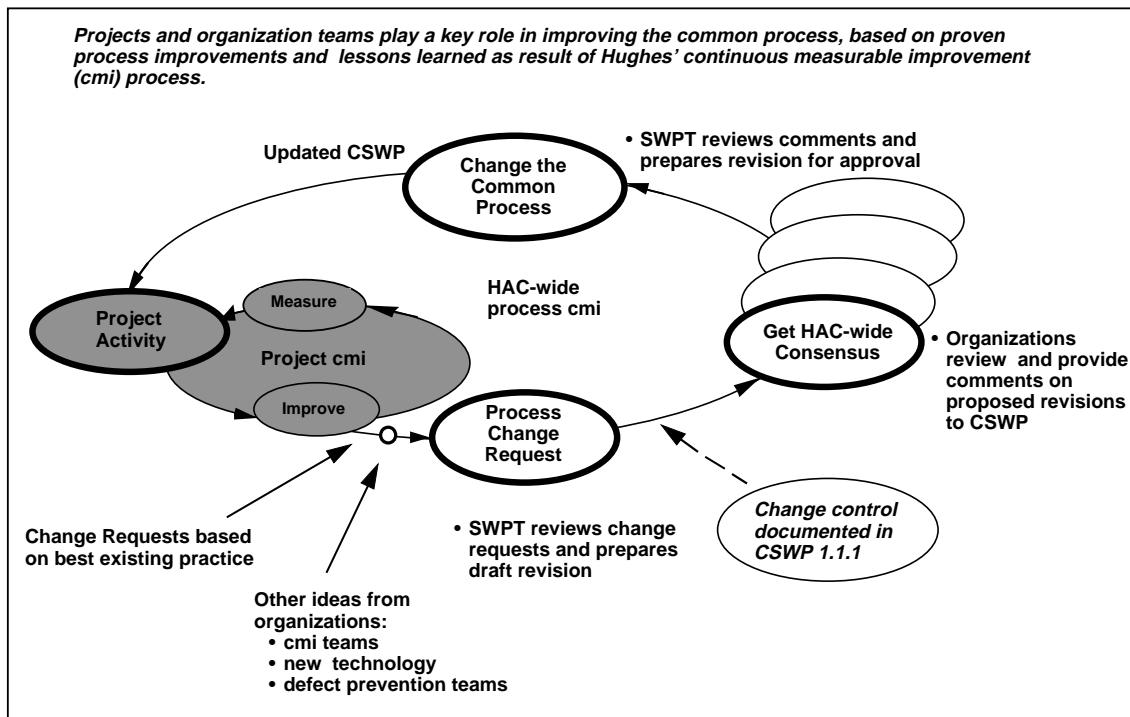


Figure 6 Change Process for the CSWP. The process provides for improvements proven on projects or piloted in organizations to be incorporated into the CSWP.

Typically, changes result from project lessons learned, and are submitted by project personnel based on process improvements incorporated into the project's tailored version of the standard process. Internal to each project, process improvements may be made and

piloted. Once proven to be successful, the improvements are documented and submitted for consideration for incorporation into the CSWP.

Changes also may be submitted as a result of processes piloted within organizations. For example, new level 4 and 5 processes are currently implemented in Command and Control Systems and are being proven on a variety of programs. With successful completion of the piloting effort, the new processes are submitted for incorporation into the CSWP and available for use across Hughes.

The SWPT reviews change requests, approves or rejects them, and oversees the complete definition of the processes with format and content compatible with the CSWP. Notification is provided back to the originator. Final approval of changes is signed by senior software process sponsors representing the major organizations throughout Hughes. Once approved and released on the Hughes Web, the updated process is deployed as described in Section 5.

The typical change process begins with an idea for an improved process, as shown in Figure 7. Untried processes are not incorporated into the CSWP; instead they are used and proven as “best practices” before they are accepted. Therefore, the process is used on one or more projects, evaluated for effectiveness, and then documented in a change request using the CSWP change request form. The form is reviewed within the submitting organization, typically by the SEPG, and submitted through the organization’s representative to the SWPT or directly to the CSWP librarian. After approval, the change may be edited for consistency with the rest of the CSWP, then distributed for participating organizations to review.

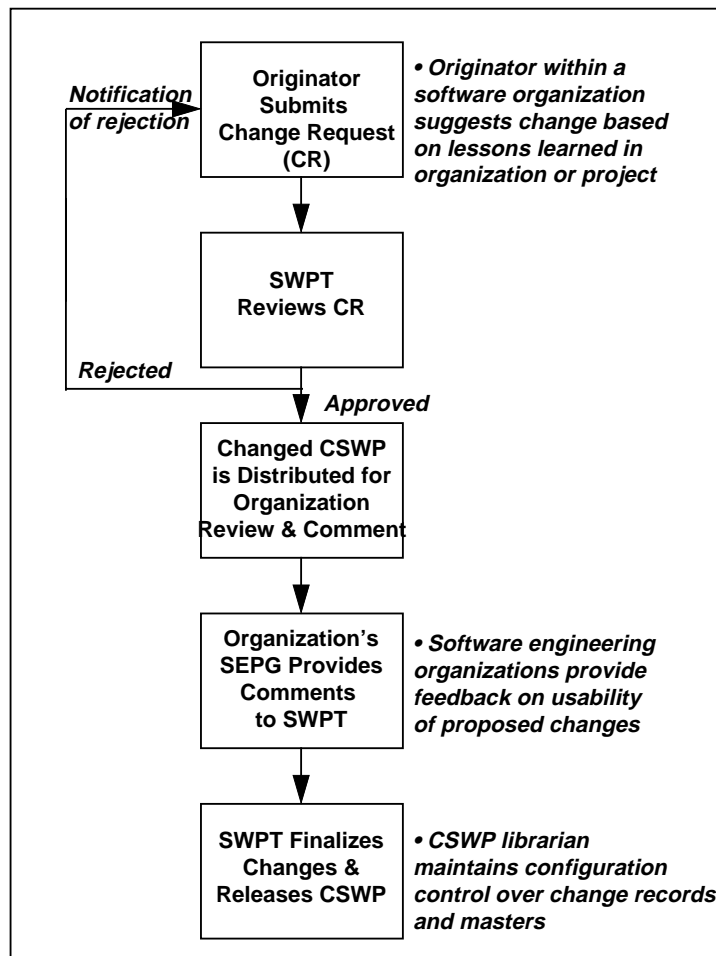


Figure 7 Change Submittal and Review Process. The CSWP librarian maintains configuration control over change records and masters.

Change requests can be submitted by anyone within Hughes, are reviewed and approved by the SWPT, and are maintained under configuration control.

Consensus is obtained by submitting the proposed change to the software organizations throughout Hughes, encouraging them to provide comments to the SWPT for incorporation into the final version prior to release. This mechanism allows all Hughes software engineering organizations to feed back comments based on their experience and helps to ensure that wording is acceptable throughout the company and that the change will meet the organizations' needs.

## Evolving Processes at Levels 4 and 5

Since the emphasis at Hughes has been on deployment of the CSWP to users across the company, it has not been a priority to implement Level 4 and 5 processes throughout the entire organization. Instead, use of levels 4 and 5 has been limited to selected divisions

within the company to ensure that the processes are mature and robust before they are added to the CSWP.

Command and Control Systems began piloting level 4 and 5 starting in 1996, with the intent of thoroughly using all aspects of the documented process before deploying it to other divisions for their use. As of late 1997, these processes have been used on a variety of large programs and are ready to incorporate into processes used in other organizations. Lessons learned on existing programs have helped to refine the documented procedures for better usability and extend them for completeness. Concurrent with submittal of the processes for incorporation into the CSWP, the written procedures and associated example documentation are made available to other Hughes organizations ready to move into the level 4 KPAs.

Feedback from the piloting provides improved processes and documentation, consistent with level 5 process change management. The CSWP change control process, itself long in use, is also part of the implementation of this KPA. Similarly, piloting technology change management provides feedback into improved technologies that can be applied in various organizations throughout Hughes.

The process of incorporation of the Level 4 and 5 processes into the CSWP is illustrated in Figure 8.

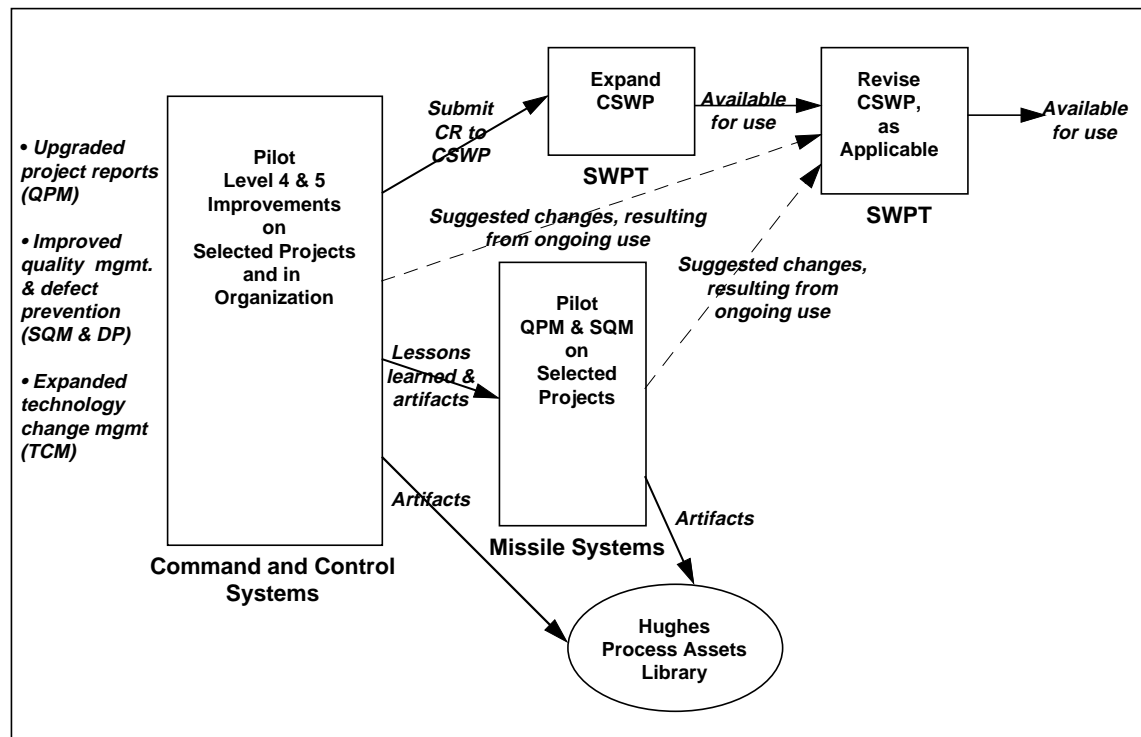


Figure 8 Process of Incorporating Level 4 and 5 Processes into the CSWP. Level 4 and 5 processes piloted and evaluated within designated Hughes organizations are incorporated into the CSWP after completion of the piloting effort.

## **Why the CSWP Has Survived and Grown**

The successes of the organizations that have used the CSWP account for its wide acceptance and growth. Prior to its adoption, some organizations were disappointed in results of SCEs and assessments. With its adoption, customer satisfaction, productivity, and product quality all have improved. In some organizations, morale increased because software engineers felt that they had more autonomy and a better understanding of their projects.

The increased use of metrics, evaluation of defects, and improvements in process as a result of peer reviews and training have had a clear impact on organizations that are newly adopting the CSWP. Both managers and software engineers feel they have a better understanding of their jobs and the status of their assignments. Peer review results provide insights into areas for improvement that otherwise might be overlooked. Benefits in cost and schedule are achieved. Recent assessment results clearly show significantly improved productivity and morale in organizations that have newly adopted the CSWP.

This history of continuing success tends to motivate others to adopt the CSWP and continue its spread throughout Hughes. In addition, strong management support for deployment of the CSWP into more organizations has stimulated the adoption of the common process throughout the entire company.

## **Assessment Results**

More than 25 software process assessments have been conducted throughout Hughes during the past 10 years. All of these assessments were conducted by qualified assessment teams, the earliest all SEI-trained, and the more recent trained by SEI-authorized lead assessors. Six of the assessments were conducted by outside firms (three by the SEI).

Continuous improvement in assessment results has occurred, as shown in Figure 9. Grouping the leading organizations (the heavy lines shown in the figure), shows the trend in process maturation throughout the company. In 1996, 80% of the Hughes software engineers who had adopted the common software process worked in level 2 or 3 organizations. Most recent assessments show that this trend is continuing, and some organizations are preparing for level 4 and 5 assessments.



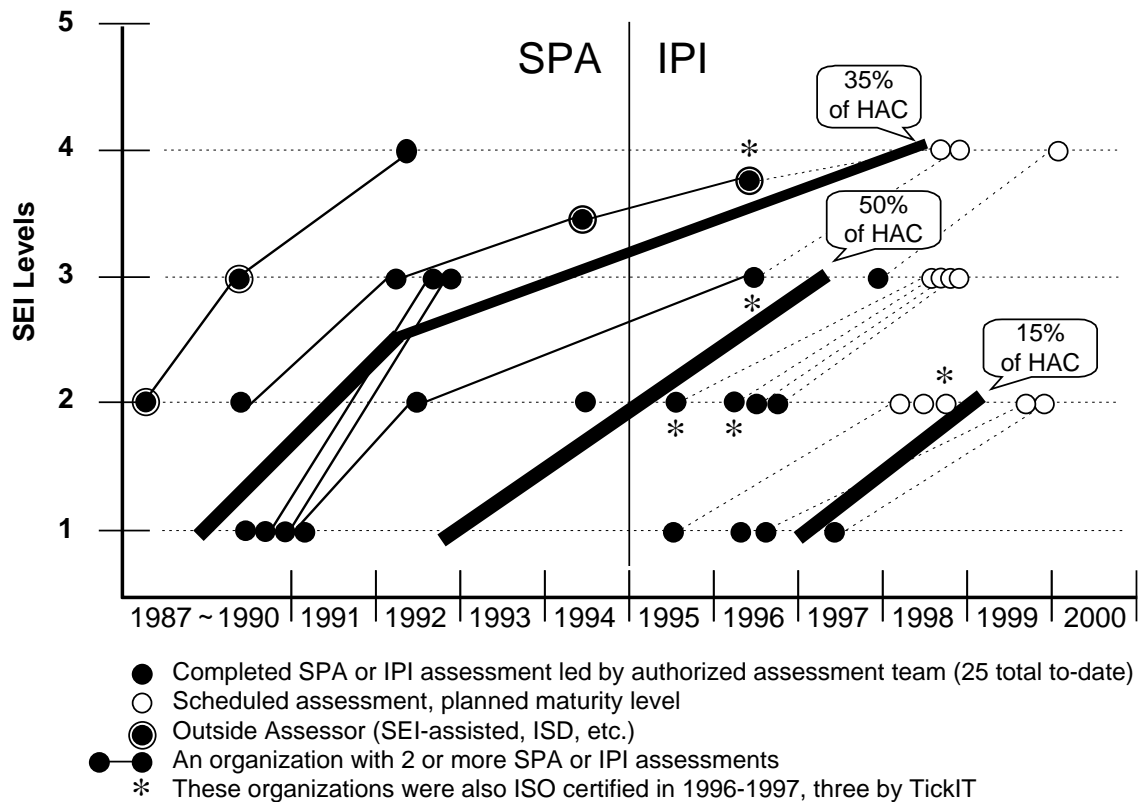


Figure 9 Ongoing Improvement in Assessment Results. Over time, assessed organizations continue to improve their processes, while new organizations undergo baseline assessments to enable them to plan their subsequent process improvement programs.

By 1998, we predict that at least 85% of the software engineers who have adopted the CSWP will be working at, or above, level 3.

At least 20 SCEs have been conducted which help validate our assessment results. Five ISO certifications (three of which were TickIT) have been awarded, and several more are in process during 1997-98. The CSWP has contributed to the success of these audits and evaluations.

Throughout the aerospace restructuring process of the 1990s, Hughes has been able to sustain the momentum toward process improvement. As the graph shows, despite reorganizations within Hughes, there is not only a sustained upward momentum, but an expanding number of organizations involved, from one division in 1987 to more than 10 in 1997.



---

## 5. Deployment of the Common Software Process Is Happening Throughout Hughes Aircraft Company

As the number of Hughes software organizations grows, SEPGs in each organization must determine the best approach to deploy the CSWP into their organizations. Prior to 1995, only five software organizations were actively involved in process improvement. In 1997, 16 organizations were actively involved in process improvement activities, and the number continues to grow. To support this growth and assist these organizations, the Hughes Process Deployment Team (PDT) was established to facilitate software process improvement across Hughes.

### Process Deployment Team

The Hughes PDT provides a company-wide forum for sharing process improvement lessons learned, collaborating on the development and dissemination of training materials, and coordinating software assessment plans.

The PDT had its origins in the early 1990s when monthly meetings were conducted to: 1) share information about software engineering environments, tools, and methods; 2) collect examples of software-related plans and artifacts; 3) provide and share training; and 4) plan software process assessments. Several teams have evolved from these early meetings, as shown in Figure 10: (1) the SWPT, which developed and provides change control for the CSWP, (2) a software tools team that evaluates new tools and provides tools recommendations and resources, and (3) the PDT.

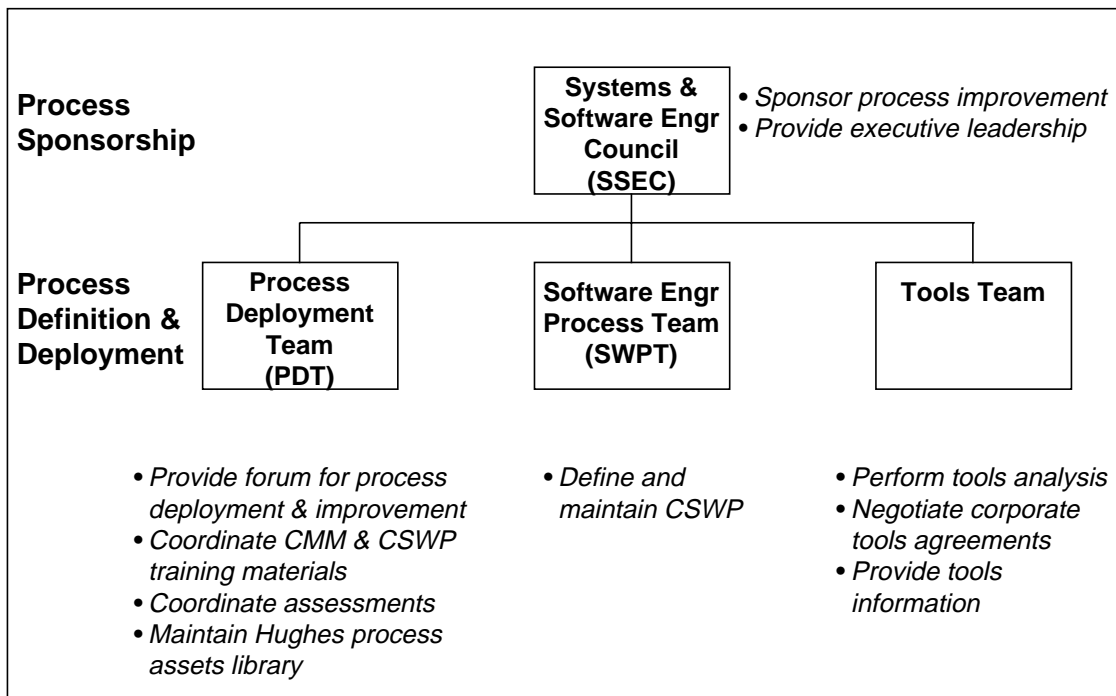


Figure 10 Teams Involved in Process Definition and Deployment. The PDT provides a Hughes-wide forum for coordination of process assets, deployment, and assessments.

The PDT creates a mechanism for software organizations' process representatives, from the various organization SEPGs, to share deployment lessons learned in a collaborative effort to raise the software process proficiency across all of Hughes. The specific vehicle for PDT communication is a series of monthly video teleconference meetings that tie together the Hughes software organizations across the nation. These are supplemented by occasional face-to-face working sessions.

Discussion at monthly meetings focuses on issues directly related to SEPG roles, such as summary of SEPG responsibilities, how to prepare a process improvement plan, establishing an organizational metrics database, guidelines on preparing for an assessment, establishing a process assets library, sharing lessons learned experiences, and similar topics.

The PDT also provides the coordination of software assessments throughout Hughes. The assessments coordinator, a member of the PDT, provides a single focal point for planning and conducting software CBA IPI assessments for all software organizations. A team of five SEI-authorized lead assessors work in conjunction with the coordinator to ensure consistency and rigor in conducting organizational assessments within Hughes. CBA IPI assessment team training is provided throughout the company under the sponsorship of the PDT.

Also, the PDT coordinates a library of process-related assets available for use throughout the company. The library is staffed by an experienced librarian who disseminates electronic and paper notices to software engineering and SEPG personnel, providing regular notification of newly received papers, documents, information about new materials, and process artifacts.

The library also contains training materials that may be shared among organizations, to provide reuse of existing materials for organizations that may wish to customize a course for their own specific needs. These include many of the course materials needed to support the twelve-step program, and other materials supporting deployment. Examples include course materials for CMM Overview, Executive/ Program Management Process Awareness, CSWP Awareness, CSWP Tailoring Workshop, KPA Completeness Workshop. One key factor associated with the process assets library is the sharing of information across organizations, which is an established part of the software engineering culture within Hughes.

## The Twelve-Step Program

The initial task of the PDT was to develop a deployment approach and the necessary resources. The approach is documented in a “twelve step program for software process improvement.” The twelve steps embody the principles of the SEI IDEAL<sup>SM</sup> model with focus on establishing process improvement sponsorship, obtaining project and line management buy-in, assessing the current state of projects, facilitating project process improvement, measuring the overall organizational effectiveness of the activities by conducting an assessment, and then establishing organization action plans.

The twelve-step program provides the framework for an organization new to process improvement to make progress quickly and establishes the foundation to achieve Capability Maturity Model (CMM) level 3.

The twelve-step process is summarized in Figure 11 and is described in the paragraphs following.

1. Establish Management Commitment and Goals.	7. Conduct CSWP Tailoring Workshops
2. Define Process Improvement Organization Roles and Responsibilities	8. Select Candidate Projects for Assessment
3. Establish Organization Process Improvement Plan	9. Conduct Project KPA Completeness Workshops
4. Identify all Projects in Organization	10. Establish and Implement Organization and Project Action Plans
5. Conduct Executive/ Program Management “Awareness” Training	11. Update Project Tailoring Reports (if applicable)
6. Conduct CSWP Awareness Training for Software Practitioners	12. Plan, Prepare, and Conduct Assessment and Identify Actions Needed for Next Improvement Cycle

*Figure 11: The Twelve-Step Program for Software Process Improvement. These steps provide a framework to guide progress in organizations new to process improvement*

1. Establish Management Commitment and Goals - The sponsorship of process improvement is critical to making progress. The sponsor is the person committed to long-term organizational improvement. This commitment minimally includes:

- organization direction, such as including process improvement in the organization goals and linking management and practitioners' incentives to these goals,
- support, such as frequent reinforcement of process goals at all levels of the organization,
- providing resources, such as personnel dedicated to process improvement tasks (e.g., SEPG), funding, and tools necessary to accomplish the associated tasks,
- establishment of process improvement tracking mechanisms.

2. Define Process Improvement Organization Roles and Responsibilities - This task includes identifying those responsible for oversight, tracking, support, and implementation. By defining and documenting responsibilities, the commitments are clearly delineated and understood by all parties.

3. Establish Organization Process Improvement Plan - This includes planning and scheduling the remaining nine steps. By completing this plan, the sponsor and the responsible parties (from step 2) analyze the twelve steps and the necessary commitments required to execute them.

4. Identify All Projects in Organization - A list is prepared of all the organization's projects, key personnel, and SEPG representatives. This allows the sponsor and SEPG to clearly understand the scope of the organization's software business, project types, and life cycle focus in order to scope the approach to their process improvement.

5. Conduct Executive/Program Management Process Awareness Training - Historically, process improvement efforts have failed if the management does not fully support the improvement activities. This training provides data on return on investment (ROI), provides an overview of the common processes, and clearly links the process improvement activities to goals, defining the potential impact of the process improvements.

6. Conduct CSWP Awareness Training for Software Practitioners - Training is provided to all personnel involved in software-related activities. This includes personnel performing support activities as well as the development community. An overview of each directive is provided, with specific focus on organization, project management, development, and software quality processes.

7. Conduct CSWP Tailoring Workshops - These relate the "new" processes from the CSWP to projects' "as-is" project directives and processes, and provides the forum to initiate new projects into full CSWP implementation. A summary of the tailoring is documented and approved.

8. Select Candidate Projects for Assessment - A subset of projects that will provide adequate organization and life-cycle coverage during an assessment is identified jointly by the SEPG and sponsor. Concurrence is obtained from the program and software project managers to ensure their cooperation. The selected projects become the primary focus for process deployment (the remaining steps).

9. Conduct Project Key Process Area (KPA) Completeness Workshops - The SEPG and candidate projects work together to develop a detailed mapping of all the CMM level 2 and 3 key practices to the existing organization and project data, including organization and project directives and evidence of implementation. Subsequently, the identified data is gathered and reviewed to ensure that it is appropriate and complete. This constitutes a self-assessment of the organization and projects. Missing processes and data are noted and used in the next step.

10. Establish and Implement Organization and Project Action Plans - Project improvements are planned to provide full compliance with the CMM. In addition, organization actions address the areas identified in the KPAs that are primarily the organization's responsibility. The action items are documented as appendixes to the organization and project process improvement plans.

11. Update Project Tailoring Reports - Projects completing their actions have typically updated their processes to provide the necessary data. As a result, the tailoring report (generated in step 7) may require revision to represent the current processes. This ensures that project tailoring reports are kept up to date.

12. Plan, Prepare, and Conduct Assessment - At this point, the organization and its projects have implemented a number of process improvements. They are now ready to plan and undergo an independent review of their processes. To prepare for the assessment, examples of project and organization data are collected and organized for review by the assessment team. As a result of the assessment, actions are identified for the next improvement cycle.

Status of the organization's process deployment efforts is periodically reported to management using a report similar to project deployment status, "the thermometer chart." Thus, both the status of the project efforts and status of SEPG-coordinated process deployment efforts are regularly tracked and presented to the sponsor and senior management, providing clear visibility into improvement plan progress, as shown in Figure 12.

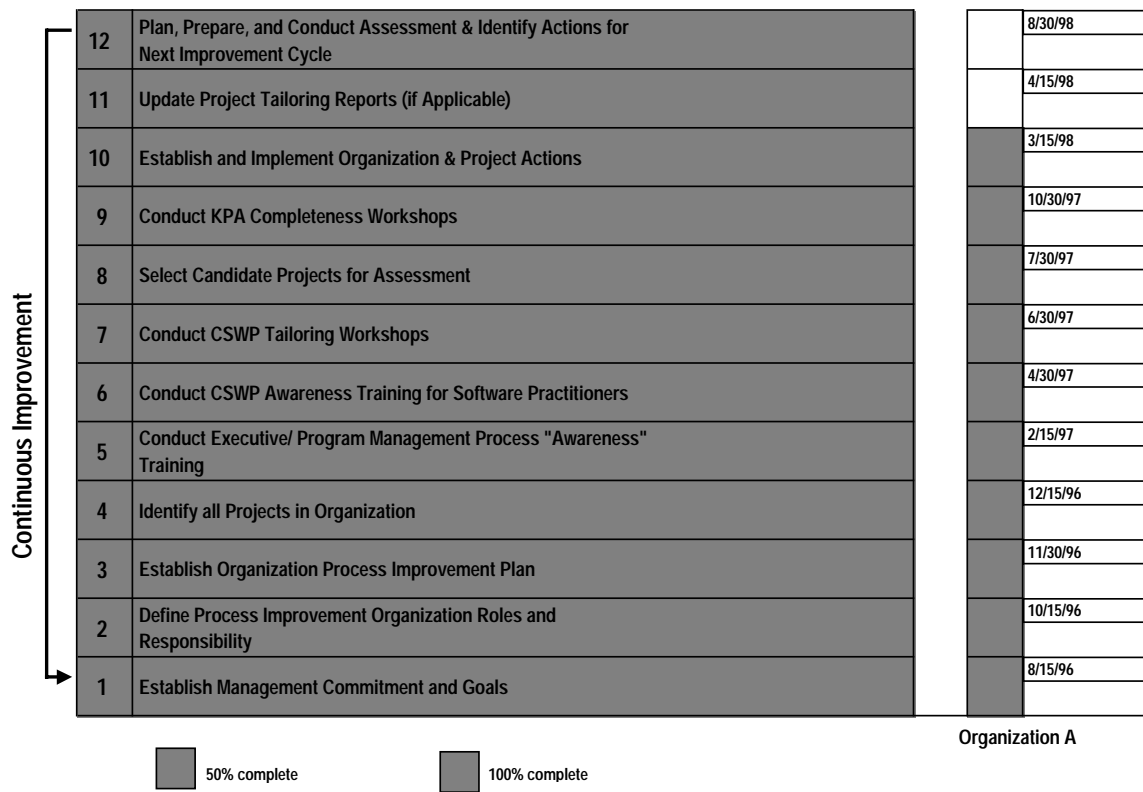


Figure 12 Status of Organization Process Improvement Activities. Status is tracked by the SEPG and reviewed with the sponsor.

Individual projects have a similar set of twelve step tasks that support the organization’s twelve step program and reflect their process deployment status. This provides a mechanism to assign process improvement responsibility to the projects, encouraging them to work in conjunction with the SEPG to implement their respective actions. Several Hughes organizations maintain status reports similar to the one shown in Figure 12.



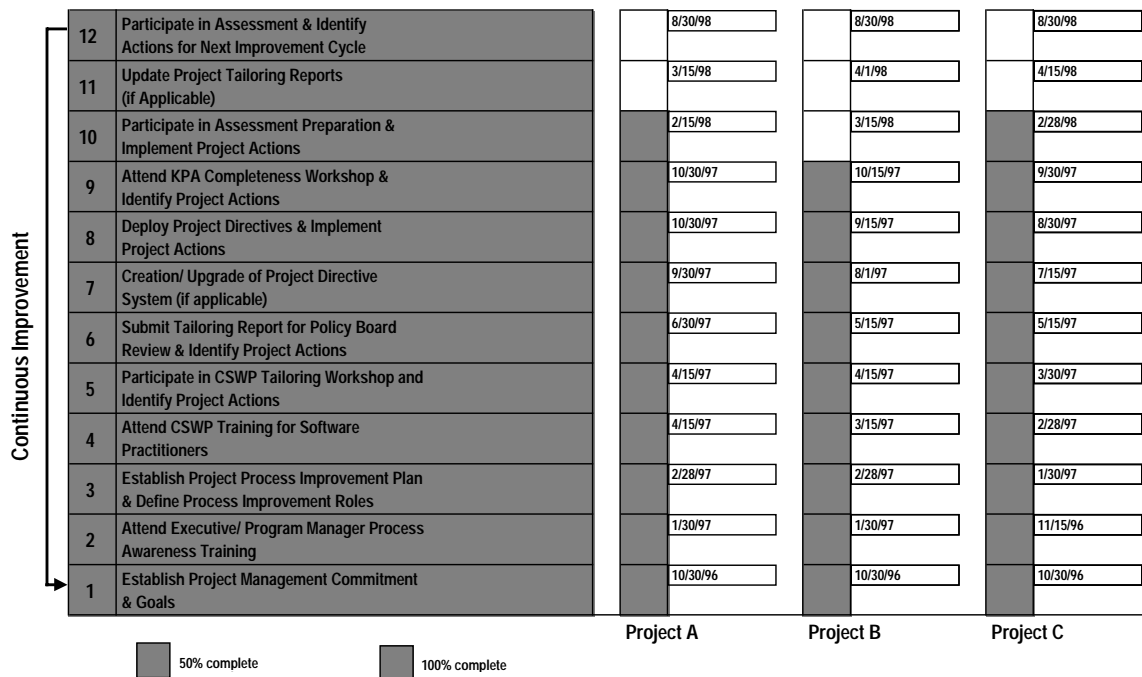


Figure 13 Status of Project Process Improvement Activities. Status is tracked and reported to the SEPG for each project, then collectively reported to the sponsor for management review.

The benefit of the twelve-step program is that it provides a process improvement “map” for organizations that clarifies the actions required to identify and implement improvements prior to an assessment. This approach is adapted for use throughout Hughes.

### Software Process Assessment Approach

Hughes has a cadre of SEI-authorized lead assessors and trained assessment team members to provide independent assessments to satisfy the CBA IPI methodology. Teams are established following strict guidelines to ensure independence and rigor in each assessment.

To provide an objective look at the organization’s processes, team composition ground rules include: the lead assessor must come from an external Hughes organization and less than half the team may come from the assessed organization. Hughes assessments also follow stringent data corroboration rules, which require explicit substantiation with evidence of implementation. This provides consistent results across Hughes and provides the organization with clear findings to support its ongoing process improvement program. This approach also supplies a clear picture of what to expect in a competitive customer-conducted SCE.

Following an assessment, the organization is responsible for establishing its next process improvement plan, incorporating actions to address assessment findings. The twelve-step

program may be adapted to meet the organization's needs, assist in planning, and accommodate new projects.

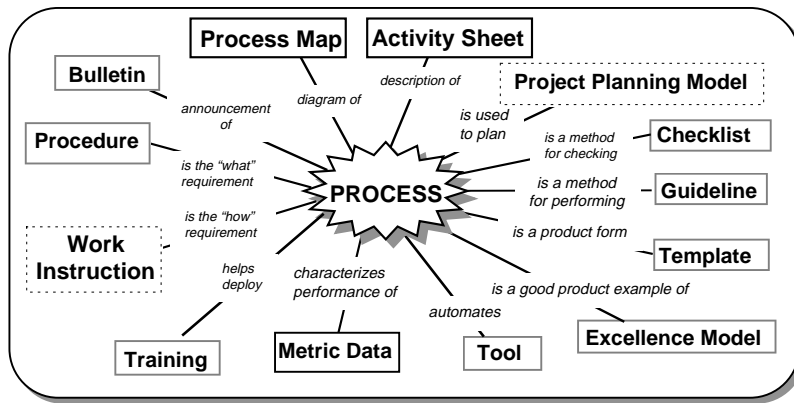
---

## 6. We Have Many Associated Assets Supporting the Common Software Process

The CSWP is part of a larger Engineering Process System (EPS) that provides a framework and repository for documenting the software and systems processes. The various SEPGs use the EPS to support their activities.

Software organizations assign SEPG members to one or more projects as their principal focus. One of their responsibilities is the deployment of process assets to the project. To effectively accomplish this, numerous process assets are available to sustain the CSWP deployment and improvement. For example, a corporate infrastructure with a process focus, training, SEPG networks, metrics databases, historical data, subject matter experts, experience, corporate improvement goals, a directive system for assessments and action plans, guidelines, checklists, project planning models, tools, and many others. The “starburst” diagram shown below is our database model for organizing and maintaining those assets. Our EPS, built upon the Hughes-wide Intranet, is used to make these assets available to the engineering community from their desktops. We put all of these assets into place to sustain a long-term deployment and improvement of the common process.

One aspect of these assets—training—is shown in Figure 14. There are 60 formal courses (many more if we include tools) available to train engineers on the CSWP. Each course comes with an abstract (shown for the Proposal and Cost Estimating course) that is available through the Hughes-wide web. The abstract can be used to call up subject matter experts for help or to schedule classes to meet the needs of periodic employee development planning.



Ada - Designing Large R-T Systems	24	Hatley-Pirbhai Overview	4
Ada - Management Overview	8	Interpersonal Communication	8
Ada 83, Ada 95, Overview	32	ISO 9001 For Software	2
Ada 83, Overview	20	Object-oriented Methods	8
Ada 95 for Ada 83 Programmers	26	Peer Reviews	2.5
Ada concepts	4	Programming - Object-oriented	22
Ada, Advanced, Generics	1	Project Management, Tools/Techniques	20
Ada, Advanced, Tasks	18	Proposals and Cost Estimating	4
Analysis and Design - Object-oriented	40	QPMIS, Introduction to	6
Assure Software Process Integrity	6	Quality in Software Design	2
C Programming	22	Quantitative Process Management	2
C Programming, Advanced	20	Radar SW System Engineering	20
C++ for C Programmers	20	Six-Sigma Concepts Applied to SW	1.5
C++ programming, object-oriented	32	Structured Design	3.5
C++, Advanced	12	SW CM for Managers	1.5
C++, Introduction	20	SW Configuration Control System	2
CMM Overview	5	SW Configuration Management, Applied	3
CSC Integration and Testing	3	<b>SW Cost and Schedule Estimation</b>	<b>16</b>
CSCI Testing	4	SW Engineering	10
CSU Testing	3	SW Engineering under J-STD-016	8
CSWP Overview	4	SW Life Cycle Processes, Alternatives	4
CSWP Tailoring Workshop	4	SW Metrics	8
Data Engineering Overview	12	SW Project Management	12
Database Design	12	SW Project Tracking and Reporting	3
Database Management Systems	12	SW Quality for Projects	4
Database Object-Oriented Design	12	SW Risk Management	4
Earned Value	20	SW Subcontract Management	3.5
Effective Meetings	8	Team Development Training	24
Fagan Inspection Technology	1	Writing Winning Proposals	40
Good Coding Practices	3	X/Motif Programming	20

TOTAL = 682 hrs

**A "Masters of Science" Degree in Software Engineering**

**Abstract**

### Proposal and Software Cost Estimating

Summary	This course introduces the student to an overview of how to prepare for and respond to Requests for Proposals (RFPs).
Coverage	Understanding the Proposal Process, Evaluating RFPs, Software Sizing, Cost Estimating, Cost Models, (e.g., SEER).
Length	6 hours
Target Audience	Engineers & managers
Prerequisites	None
Skills/ Proficiency	Entry level knowledge of working on a proposal team and performing software cost estimating.
Learning Objectives	1) Understand an RFP and the proposal process 2) Understanding of interdependencies in the process 3) How to tailor the process for a proposal effort 4) How to do software cost estimating 5) How to link the proposal process to project planning
Major Topics	1) Overview of Process 2) The Pre-RFP Process 3) Responding to a Request for Proposal (RFP) 4) Software Cost Estimating 5) The Post Proposal Submittal Process 6) Transition from a Proposal to Project Startup
Delivery Format	Presentation
Classroom Setup	Course notes, Overhead Projector
Comments	Designed as a Just-In-Time course, to be taken at initiation of the proposal or upon award of the contract.
Who to Contact	Hughes Institute For Professional Development Advanced Technical Education Program (ATEP) (310) 568-7481
Subject Matter Experts	Maryalice Toby, Jean Whitaker, Margot Harris

Figure 14 Training Assets Support the Common Software Process

Engineering managers use Hughes Employee Development Process to develop individual training plans for their employees. The Hughes Employee Development is integrated into an employee's performance appraisals. Managers and employees plan out their goals for required and "growth" training on an annual basis. Training coordinators manage the training courses offered to ensure that the appropriate training is available to meet the employees' needs. Employees provide their training records to the training coordinators, who track the training completion status.

---

## 7. The Cornerstone of Our Process Is the Program Review

The program review is the cornerstone of Hughes Aircraft's CSWP. It enforces and encourages our common process theme of strong management sponsorship and oversight, common metrics, communication, and management support.

Senior management sponsorship is at the heart of any effective process improvement program. For Hughes, this sponsorship goes beyond the supply of dollars and personnel. It extends to the active participation of senior management in the overall software process. An important avenue of participation is the program review.

The purpose of the program review is varied. It provides:

- senior management with sufficient tracking and oversight to make critical program and organizational decisions and allows them to assess program health
- the software program manager (SPM) the opportunity to request assistance and support from senior management
- a communication forum to share lessons learned among programs
- a monitoring mechanism to ensure the establishment and deployment of common metrics and reporting
- support process improvement within the organization and the corporation

Attendees at the program reviews typically include functional senior management, the SEPG, Software Quality Assurance (SQA), and the SPMs. Program Management is also invited.

Generally speaking, the details of the program review are not as important as the "culture" it nurtures within an organization. Culture-building is accomplished through active and consistent senior management involvement and support.

Involvement and support also acts as a catalyst for SPMs. SPMs assign a higher priority to the establishment and use of metrics when they know that management is showing an interest in the program and is monitoring progress.

Of the 85% of the population at Hughes Aircraft Company who have adopted the CSWP, more than half are doing program reviews, although many are still in the process of institutionalizing it. Those who have implemented have done so in a variety of ways. These variations essentially fall into two categories. Both categories dictate that the SPM prepares a Monthly Program Review Package (MPRP) that consists of the reports and metrics prescribed in the CSWP, or a subset thereof as defined in their Program Tailoring Report. The CSWP defines a total of 20 such reports and metrics. These reports and metrics are outlined in Table 4. Data shows that the initial set-up of the reports (done as part of program planning) takes between 20 and 40 hours depending on the person's experience with the reports. Once initiated, data collection and report generation typically takes between 2 and 4 hours monthly. During the conduct of the review (in both methods), key issues are identified, action items are taken and assigned, and old actions are reviewed and given a status. Each month's data is saved and used in subsequent months to compare plans versus actuals, and as historical data.

<b>Report</b>	<b>Description</b>
Project Overview	Provides summary information about the program and the contract. It also includes a system configuration diagram and the master program schedule.
Accomplishment Summary Report	Used to emphasize significant activities or accomplishments made by the program during the current reporting period.
Problem Summary Report	Summarizes the problems that require management cognizance and proposed solutions.
Project Schedule	Depict a gantt chart form of the detailed software tasks to be performed on the program.
Risk Status Report	Contains all the software program risks identified and tracked on the program. It includes the risk description, mitigation plans and current status.
Milestone Report	Provides a tabular list of all milestones and deliverables, their due dates, completion dates, and reasons for any slips.
Rate Chart Report	Used as a management tool for monitoring and reporting progress of work planned and accomplished.
Earned Value Report	Depicts the performance of work accomplished against work planned and actual expenditures in dollars and percent of the total budget.
Target System Resource Usage Report	Used to track or monitor critical system resources such as throughput, memory utilization, mass storage, etc.
Organizational Resource Forecast Report	Used to forecast the resource needs of the program as well as the resources that are becoming available. The purpose is to provide senior management insight into all resources for more effective planning and resource use.
Financial/Staffing Report	Depicts the cumulative budget, current operating plan, and actuals in both dollars and staffing.
Quality Indicator Report	Provides the defect data associated with each product produced and the analysis of that data to affect process or product improvement.
Scope Change Report	Provides management with information about how scope changes are affecting cost, schedule, quality, or technical aspects of the program. It also provides historical data.
Lessons Learned Report	Used to collect lessons learned from the program, both positive and negative, and their effects on the program. These are collected quarterly and distributed to all project leaders.
Software Problem Status Report	Depicts cumulative opened, resolved, and closed change requests over time. It is a measure of the maturity of the overall software product and a predictor of remaining work.
Productivity Measurement Report	Provides data for evaluating program performance and aids in program management, establishment of a baseline for measuring potential improvements, and collection of historical data for bidding purposes.
Software Size Trend Report	Provides insight into the growth of the program over time throughout the development life cycle.
Defect Density Tracking Report	Provides information about the quality of the products produced and the processes followed so that control and improvement can be exercised.
Software Requirements Volatility Report	Provides a measure of the stability of the software requirements.
Software Management Effectiveness Report	Provides senior management with visibility into how much management is required on a program as well as data to support replanning and future planning efforts.

*Table 4 Program Reporting, proven to be one of the most important ingredients in maturity improvement.*

In the first method, every SPM presents MPRP data. Each SPM is allocated between 30 and 45 minutes; if follow-up sessions are needed, they are scheduled. The total hours expended for this method is dependent on the number of programs (e.g., one organization held program reviews for 2 full days for 20 programs). In the second method, a subset of the SPMs present their MPRP. This typically rotates month to month, but always includes those programs that may be having difficulty. All SPMs are required to attend the subset presentations. The total hours expended for the second method is dependent on the number of SPMs in the organization and the number of programs that present (e.g., one organization limits their program review to a half day with four programs presenting).

The metrics and reports listed in Table 4 are NOT produced solely for senior management oversight. They are also used by the SPM to manage program activities and to report program status to the Program Management Office and the customer. This results in quadrupled use of the same data at the program level. In addition, this same data is rolled up organizationally and is reported to directorate or laboratory management, to the corporate process council (SSEC), and in some organizations to company management. Figure 15 depicts the typical flow of the reporting data through a program and an organization.

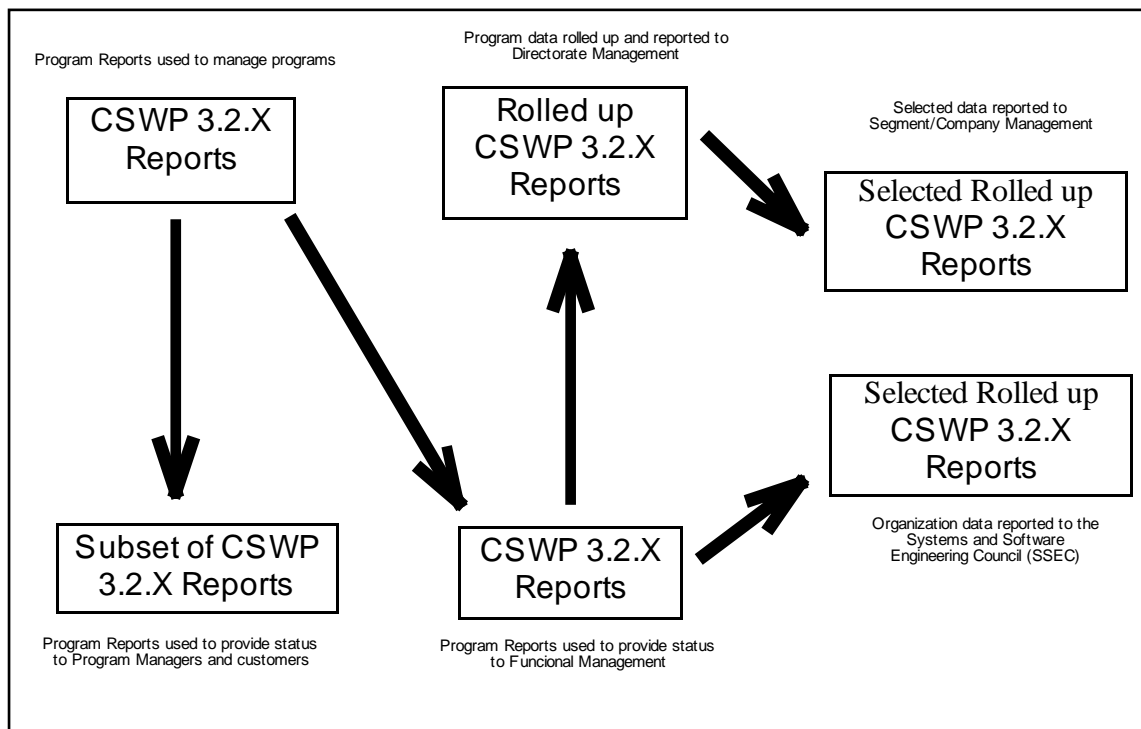


Figure 15 Program Reports provide a variety of individuals with data to support a variety of needs, both within the program and in the organization.



A typical implementation comes from the Tucson, Arizona, site. This organization, Hughes Missile System Company, adopted the corporate CSWP beginning in early 1994. They have deployed the process throughout the organization and in October 1996 achieved maturity level 3.

Program reviews in Tucson follow the first method, defined previously, and have every SPM present their MPRP, but reviews are also supported by two additional groups: the Metrics Action Team, (MAT) and the Team-of-Four (ToF).

The MAT consists of the functional managers and the SEPG. They meet once a month following the program reviews to go over the reports from all the programs as a whole. Their purpose is to step back and do an independent analysis of the data in the reports to uncover additional program key issues and to look for systemic issues affecting the overall organization. The output of this team is an updated key issues report that is used by the ToF and the program review. Systemic issues are analyzed and appropriate actions are taken (e.g., process action teams formed, directives written, directives changed or eliminated).

The ToF grew out of the need of the SPMs for support, mentoring, and counsel. In the ToF, each SPM is supported by an assigned functional department manager, an SEPG member, and the SQA representative on the program. Figure 16 depicts the ToF concept and the responsibilities of each member to the team. This ToF meets once a month just prior to the program review to discuss the issues and to support the needs of the SPM. The draft metrics and reports for the program review are also reviewed and categorized. This provides the proper level of management attention to each key issue, which is elevated to the program review. The ToF members are available to the SPM for support or counsel at any time.

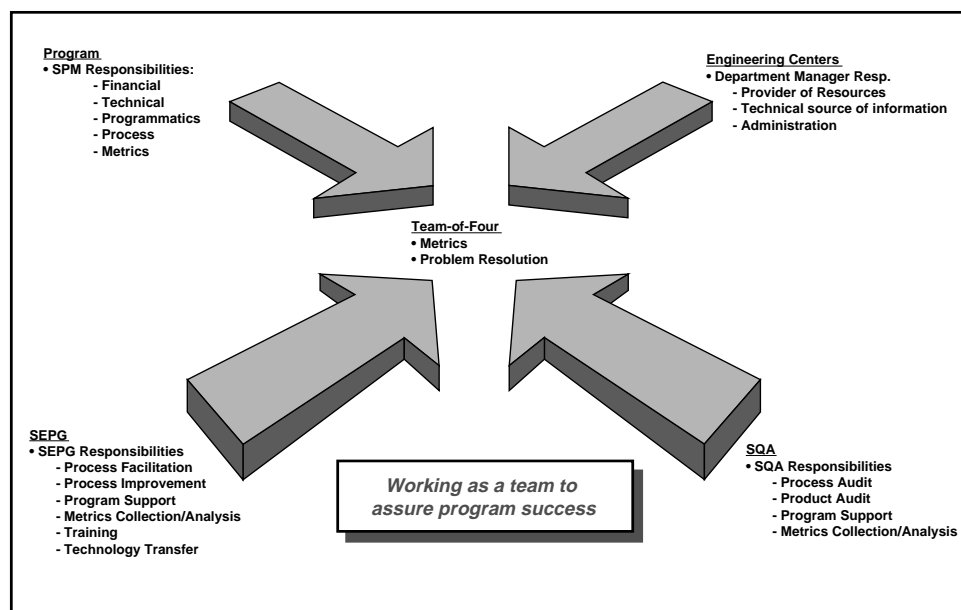


Figure 16 The Team-of-Four concept provides the support mechanism needed to help ensure program success.

The ToF process is finding its way into other Hughes organizations and is rapidly becoming another common process. This process will be offered to the Software Process Oversight Team for consideration as an addition to the common process. This is an example of how best practices evolve within the organization, are tried and proven, adopted by others, and then turned into common process.

Figure 17 depicts a typical monthly scheduling of the MAT, the ToF, and the program reviews.

Mon.	Tues.	Wed.	Thur.	Fri.
		MAT		
Teams-of-Four			Program Review	
Teams-of-Four			Program Review	

*Figure 17 The MAT, ToF, and program review provide the SPM with several opportunities for visibility and support each month.*

Figures 18, 19, 20, and 21 illustrate some of the roll-up reports that result from the program reviews and help an organization to deploy process, and monitor organizational progress and improvement, and help to provide focused program support.

Finally, the program review provides a forum to discuss potential and actual process improvement opportunities. The improvement activities happening at the program level are shared with other SPMs, senior management, and the SEPG. These improvement opportunities are then prototyped, measured, and—if found successful—added to the common process. Once proven locally, the improvements are passed to the corporate process team to be shared throughout Hughes Aircraft Company.

The program review has become a culturally accepted practice throughout Hughes Aircraft Company and has led to better managed programs. The implementations vary from organization to organization, but the results are the same. Senior management participation in the process leads to easier process deployment, organizational support and buy-in, and a more effective and efficient organization.

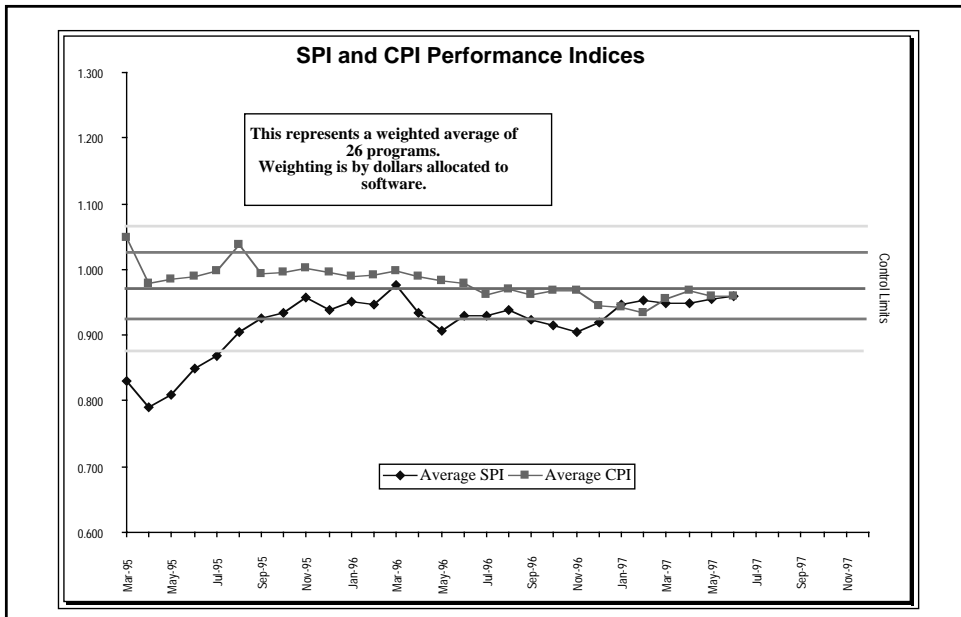


Figure 18 Organizational Schedule Performance and Cost Performance Indices indicate how the organization is performing against schedule and financial plans. This organization has been stabilizing and improving over the past two years.

**Organizational Metric History  
July 1997**

Report	Phalanx	Phalanx PASS	RAM GMLS	RAM IRMU	SEPG	SM-2 Blk IVA IR	SM-2 Blk IVA RR	SM Land Strike	SM-3	Tomahawk Block 3	Singer Block II	TBIP
Project Overview												
Project Accomplishment Summary												
Problem Summary		N/A		N/A								
Project Schedule												
Risk Status			N/A		D							
Milestone					D							
Rate Chart												
Earned Value												
Target System Resource Usage					D							
Organizational Resource Forecast		N/A			N/A							
Finance/Manpower												
Quality Indicator								N/A				
Scope Change	N/A		N/A		N/A							
Lessons Learned		N/A	N/A					N/A				
Software Problem Status								N/A				
Productivity Measurement										D		
Software Size Trend					N/A							
Defect Density					N/A			N/A				
Requirements Volatility		N/A			N/A			N/A	N/A	N/A		
Management Effectiveness	D		D	N/A	N/A			N/A	N/A	N/A		
<b>Report Totals</b>	20	20	20	20	20	20	20	18	20	13	20	20

Legend: R = Red, Y = Yellow, G = Green, B = Blue  
 N/A = not appropriate at this time  
 Blank = not reported  
 D = deleted for maintenance phase

Figure 19 Getting the status of the content of each report and metric helps to focus management attention on the key issues of each program.

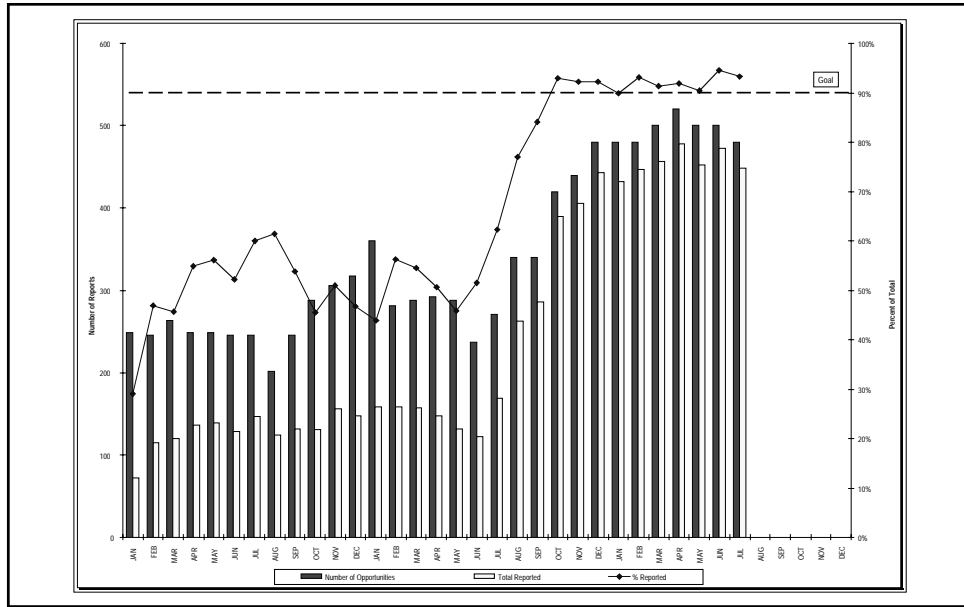


Figure 20 Tracking how well the organization is doing in the collection and reporting of data is a reflection on the deployment of the process. Note the significant increase during the past eight months. This was a direct result of the establishment of the MAT.

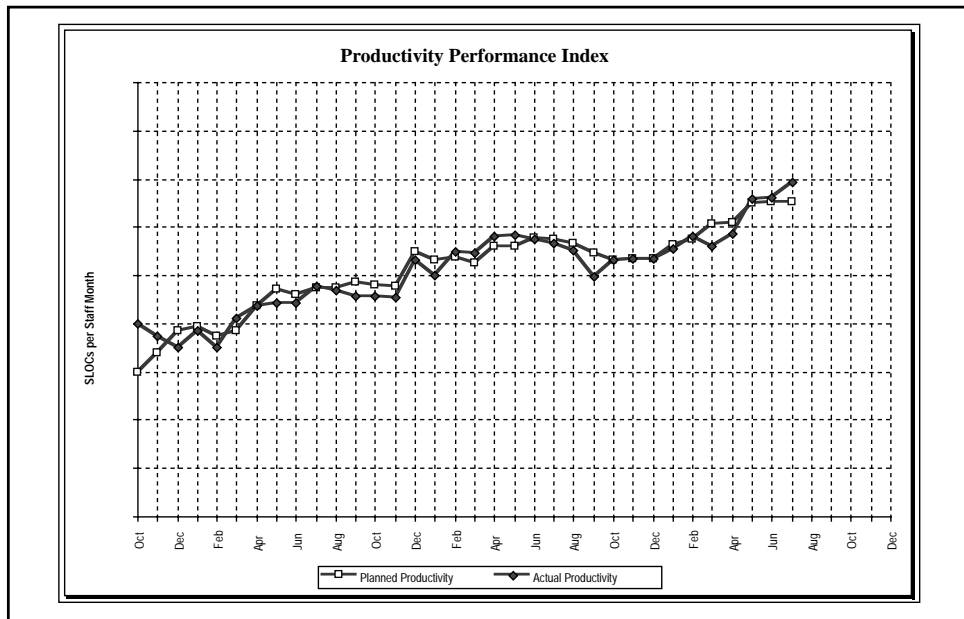


Figure 21 Productivity Measurement provides the bottom line to our improvement activities.



---

## 8. Performance Improved After Institutionalizing the Common Software Process

This section describes some of the significant performance improvements from various areas of Hughes Aircraft Company that have resulted from the utilization of the Common Software Process.

### **The CSWP Helped Achieve CPI = 1.0 Within a Small Variation**

CPI and SPI are measures of a project's performance to cost and schedule plans. They are valuable program management tools and they are important to a software organization trying to optimize its process because they indicate how well the process meets project commitments.

This write-up refers to an activity known as the "standard task" as an example. At Hughes, the standard task consists of software design, coding, unit test, and software integration.

There are three variables that must be known to calculate CPI and SPI as follows:

**Earned value** is a measure, expressed in dollars, of how much work has been accomplished at a given point in time. There are several ways that earned value systems can be set up. The following method is used to determine earned value for the standard task:

- Milestones for completion of design, code, unit test and software integration of each module (~50 lines of code each) are established during planning. Milestones are weighted based on complexity and type of software.
- At each status interval, the percent of these weighted milestones that have been completed is determined.
- Earned value is equal to the percent of weighted milestones actually completed times the budget for the standard task.

**Budgeted Cost of Work Scheduled (BCWS)** for the standard task is equal to the percent of weighted milestones planned to be completed times the budget for the standard task. It measures the value of work scheduled to be complete at a given point in the schedule.

**Actual Cost of Work Performed (ACWP)** for the standard task is the actual cumulative cost incurred for the work completed.

**CPI and SPI** are calculated as follows:

$$[1] \text{ CPI} = \frac{\text{Earned Value}}{\text{Actual Cost of Work Performed (ACWP)}}$$

$$[2] \text{ SPI} = \frac{\text{Earned Value}}{\text{Budgeted Cost of Work Scheduled}}$$

CPI is a comparison of the value of what has been completed versus what it has cost and thus indicates cost performance.

SPI is a comparison of the value of what has been completed versus the value of what was scheduled and thus indicates schedule performance.

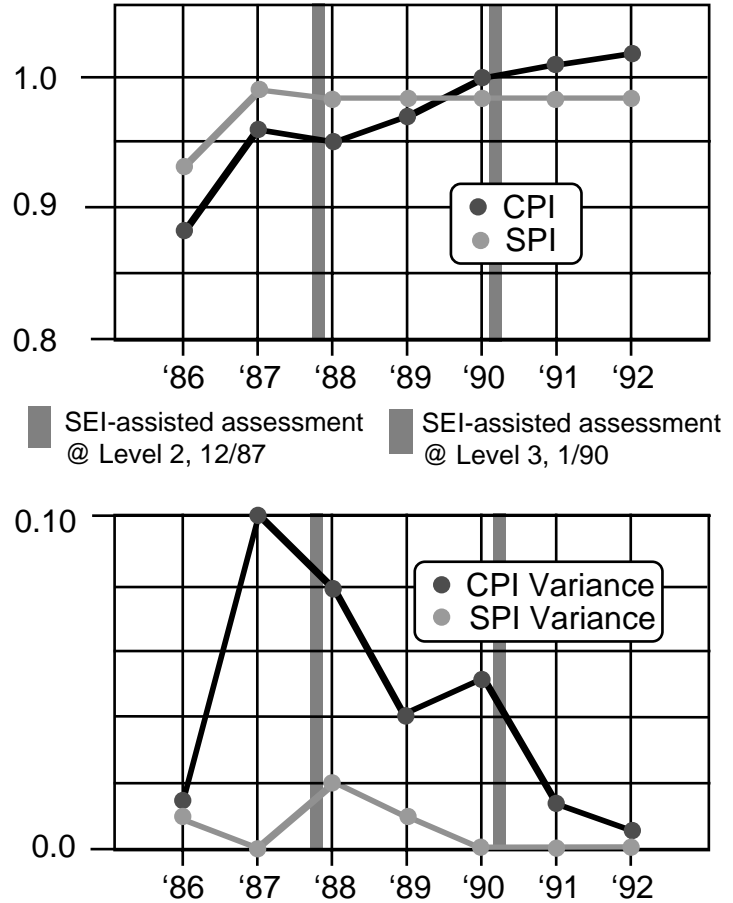
If the CPI is 1.0, the project is meeting its cost objectives. If the CPI is greater than 1.0, the project is underrunning its cost. Similarly, the meaning of an SPI of 1.0 is that the project is earning value at the rate planned—i.e., on schedule. If the SPI is higher than 1.0, the project is ahead of schedule.

An earned value system can be set up for any task with identifiable products to which a value can be assigned. Other software tasks are also added into the software earned value and thus the CPI and SPI calculations.



Figure 22 CPI and SPI were optimized by process improvements during the early 1980s and following SEI-assisted assessments in 1987 and 1990.

Starting in the early 1980s, we began using CPI and SPI to diagnose our common software process. Many improvements were made to the process, but most important were the improvements caused by the SEI-assisted assessment in 1987. Since we had been collecting this data all along, average CPI and SPI values before, during, and after the two SEI-assisted assessments gave us insight into the effect of the process maturity improvement of our common process from level 2 to level 3. The gain in CPI saved the organization about \$2M per year. The average CPI and SPI during those years are plotted in Figure 22. Also, the variance in the average CPI and SPI is plotted, showing that the average moved toward the desired value of 1.0 and the variance was reduced.



The CSWP was changed to capture these improvements and the process has been continuously improved to bring CPI and SPI closer to the desired  $1.0 \pm d$  for some small  $d$  (delta).

## Higher Maturity Yields Better CPI

Figure 23 shows that process maturity improvement pays off. For each software project represented in our database we have captured project-end CPI value and estimated process maturity level (many maturity levels are known as they were included in formal assessments). Each “+” represents one project’s CPI value correlated to its maturity level. This figure shows that, as process maturity improves, CPI values cluster around the target value of 1.0 and the variance gets smaller. That means better predictability on contracts and fewer cost overruns. This data from our database of software projects matches closely the data reported in the fall 1996 *IEEE Software Process Newsletter*.

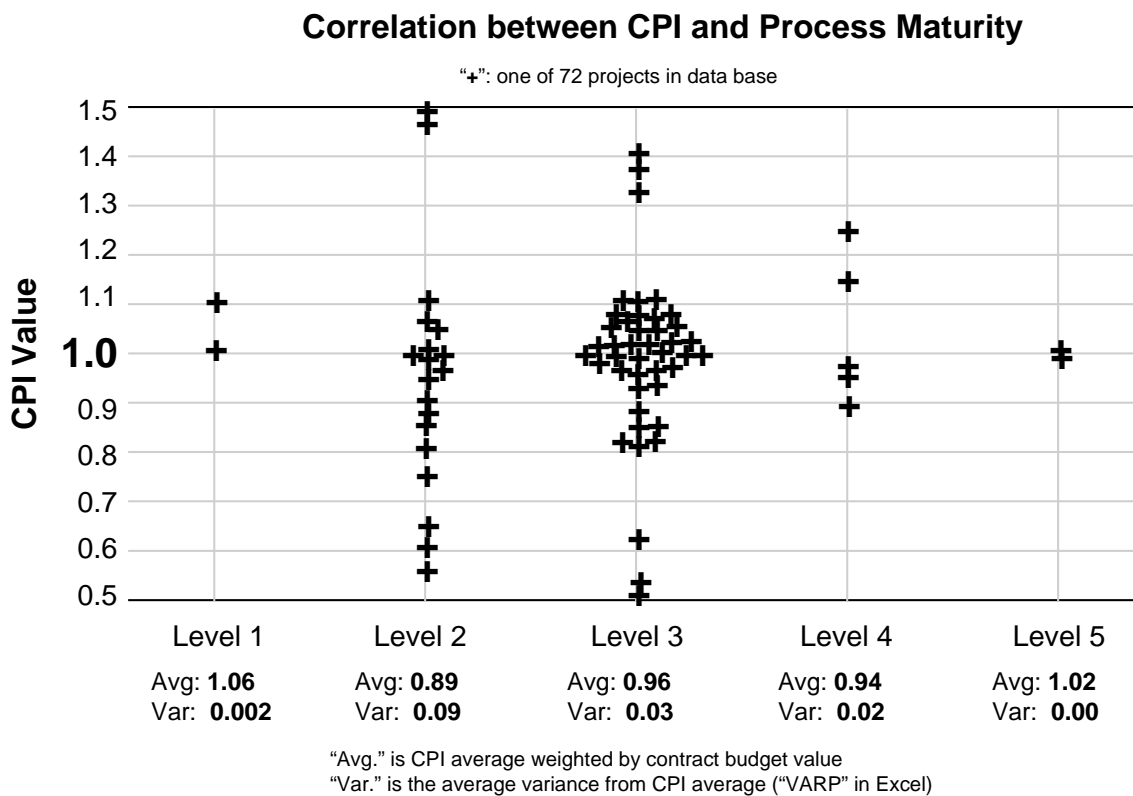


Figure 23 Correlation between CPI and Process Maturity. As process maturity improves, CPI values cluster around 1.0 with minimal variance.

## CPI/SPI History for the Peace Shield Project

The highly successful \$1.2 billion Peace Shield contract illustrates the predictability of our software process based on the CPI and SPI indices. Peace Shield was planned to the 51 month bonus award schedule (rather than the 54 month contract schedule) and was managed to that date. As shown in Figure 24, actual completion was 48 months—6 months ahead of contract requirements—because of continuous optimization of the software process. SPI, as well as CPI, was 0.99. As a result, \$50,000,000 were returned to Hughes as a performance bonus by the customer. Peace Shield was featured in the May 29, 1995, issue of *Aviation Week and Space Technology*.

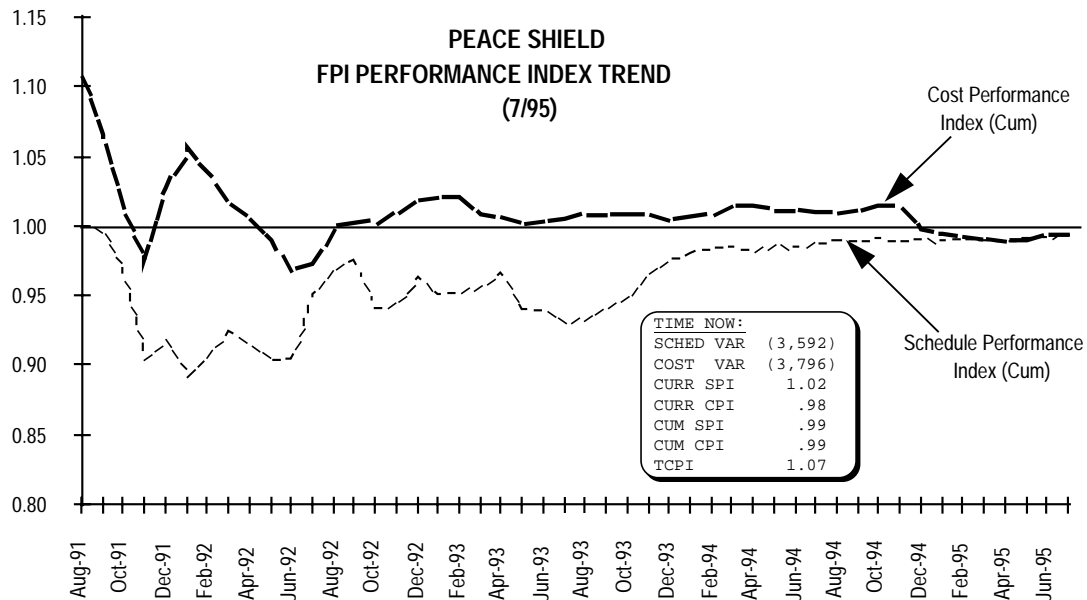


Figure 24 CPI and SPI Performance on Peace Shield. Predictability and continuous optimization of the software process resulted in a \$50,000,000 performance bonus.

## Review Efficiency Improved

We implemented a process improvement in 1991 based on defect prevention analyses. We noted from our data that only 43% of the requirement product defects were being found during the phase in which they were created. To improve the process, we decided to focus more attention on requirements reviews and expand the requirements review checklist to spell out exactly what is expected for a high-quality requirement. The graph in Figure 25 summarizes a database of 66,584 defects found during reviews and test that were detected, classified, and stored in our Quality Indicator Program (QIP) database for all projects ending during the years 1989 to 1996 (only completed projects can be used; otherwise the percentages within each phase will not be accurate).

We divided the database into two databases: defects from projects starting before the improvement (1989 or 1990) and projects starting after the improvement (1991 and later).

For the “before” database, the projects would not have taken advantage of the improvement, whereas projects contributing to the “after” database would reflect the change caused by the improvement.

The “before” database classifies 38,120 defects, 1702 of which are requirements product defects. The “after” database classifies 28,464 defects, 3038 of which are requirements product defects. Based on the process improvement in 1991, the percentage of requirements defects caught in phase improved from 43.4% to 84.5% as illustrated in the graph for the two bars in the “Req Anal.” column. With no process improvement, only  $3038 * 43.4\% = 1318$  defects would have been found in phase rather than the actual  $3038 * 84.5\% = 2567$ . Thus, we did *not* need to find and fix  $2567-1318=1249$  latent requirements defects. The improved process was then implemented as a change to the common process so that all future projects using the process would also realize similar savings. This type of analysis is performed for each phase’s review efficiency to ensure that we have continuous process improvement.

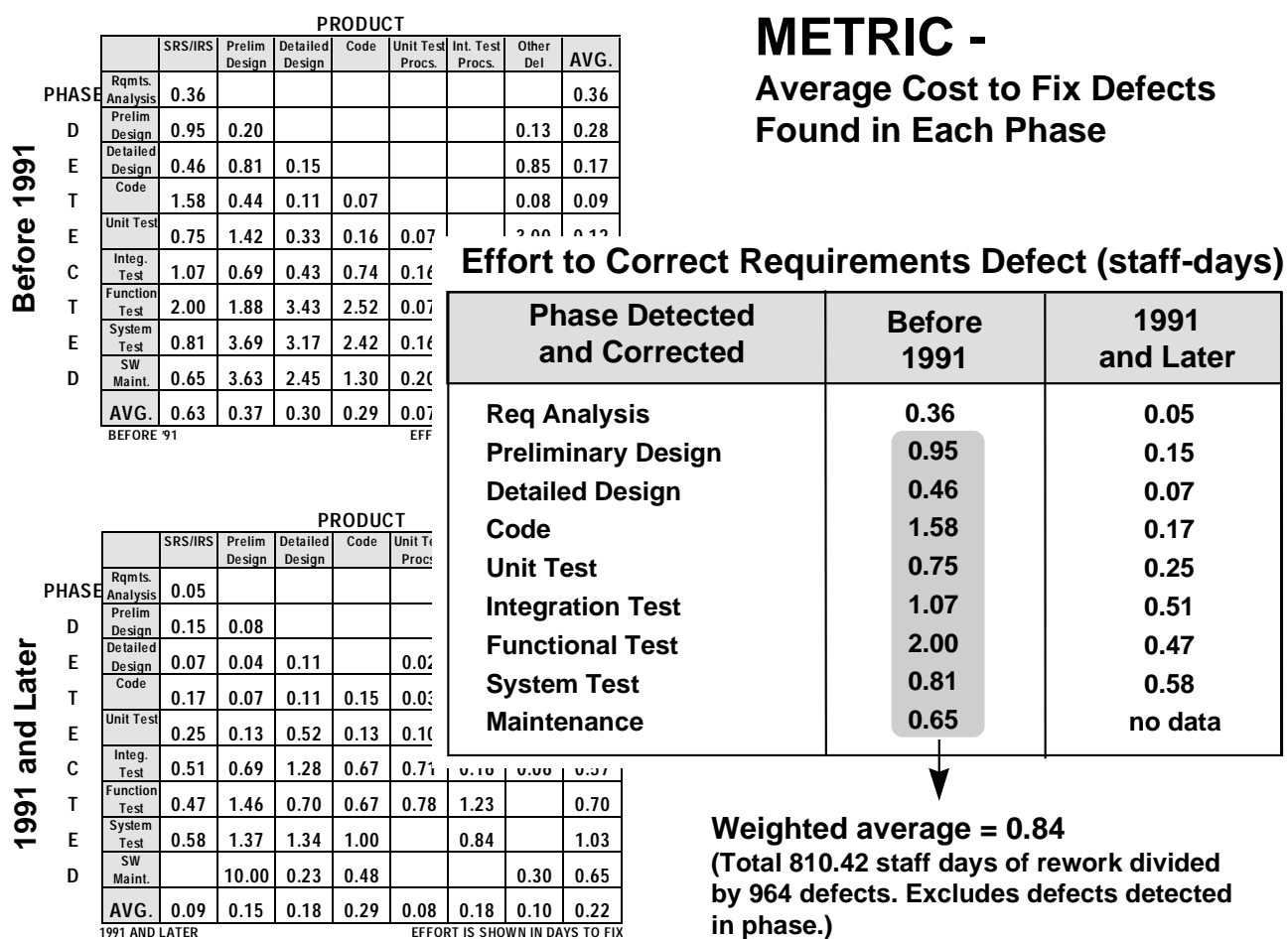


Figure 25 Analysis of Quality Indicator data resulted in requirements review efficiency improvements that reduced overall effort.

## Productivity Improvement of 987 Staff-Days

The matrices shown provide actual data reported by QIP from the same database as for the review efficiency improvement discussed above. They report average effort in staff-days (sd) needed to fix defects. When a defect is detected (phase detected), its type is classified and the effort in staff days to fix the defect is captured. The detailed matrices show the average cost to fix defects of each product type as discovered in each software development phase. The shaded boxes above each column indicate the products in which the defects were discovered. The shaded boxes at the left of each row indicate the development phases in which the defects were found. The far right column in each matrix shows the average cost for defects found in each corresponding development phase. The bottom rows show the average cost for defects of each product type.

The large table to the right summarizes only the requirements defect costs for the “Before 1991” and “1991 and Later” data sets. The rows in the table show development phases; the columns give the average effort to correct a requirements product defect within the corresponding development phase. For example, the effort to fix a software requirements specification defect in the code phase is 1.58 staff days, shown at the intersection of the “Before 1991” column and the “Code” row. When a defect in a requirements product is detected in the Requirements Analysis phase, the cost to fix that defect is less than detecting it and correcting it in a subsequent phase. This can be seen in the detailed tables. One reason for this is that a single requirement (defect) may be allocated to several design components which, in turn, become “contaminated” by the host defect. Now, to correct the single requirements defect means that several design components must be corrected as well.

As is evident from the data above, correcting a requirements defect within phase in the “1991 and Later” database cost only 0.05 staff days, seven times less than the 0.36 staff days in the “Before 1991” database. The weighted average effort for latent defects in the “Before 1991” database is .84 staff days. Recalling the 1249 latent requirements defects from the review efficiency improvement shown in Topic 9, those 1249 latent defects would have cost us  $1249 * .84 \text{ sd} = 1049 \text{ sd}$ . Instead, the cost of fixing the defects in phase was  $1249 * 0.05 = 62$ . Thus, we saved  $1049 - 62 = 987$  staff days for the projects starting after 1991.

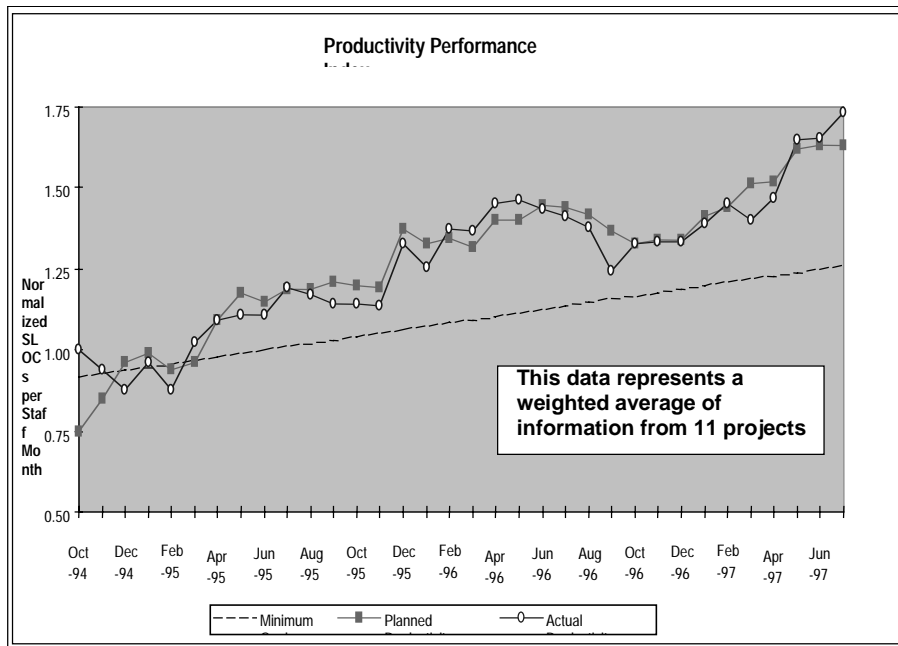


Figure 26 Five-Year Productivity Improvement Goal Achieved in Three Years. Process maturity increased by one level during the performance period.

Other evidence of productivity improvement is shown by Figure 26. This recent performance profile is from a major Hughes organization that increased its process maturity rating by one level during the performance period. The organization had established a minimum goal necessary to achieve a 50% productivity improvement over a five year period. They were able to achieve the improvement in just under three years.

An interesting debate has arisen as to when finished projects should be removed from the database in graphing future productivity. The information was weighted by the number of source lines of code for each project in determining the average productivity. A heavily weighted project can cause significant spikes or dips if the sample size is small and if the project is more productive or less productive than the average. Our initial thoughts lean toward including the data for one to two years after completion.

The data presented in this section have shown substantial performance improvements in terms of costs, schedules, and quality as a result of the institutionalization of the CSWP. Continuous measurable improvement activities will ensure future performance gains.

---

## 9. Our Customers Are Pleased

After all is said and done about the effectiveness of the CSWP—including the standard project reporting, our defined metrics, our commitment to process improvement, the widespread deployment of the process and so on—the real importance of the process improvement effort is that we have satisfied our customers. This section gives specific quantitative examples about three ways in which we have pleased customers across the board—improvements in cost, schedule, and quality—and then illustrates some direct feedback from two pleased customers.

### Improved Cost, Schedule, and Quality on Programs

A clear-cut quantitative benefit of improved processes to our customers is our improvement in cost and schedule performance indexes, CPI and SPI. As discussed earlier in the section on performance, and as shown in the top portion of Figure 22, the average CPI for all projects in SED during the years 1988 to 1990—the time period when the process maturity improved from level 2 to level 3—improved from CPI of 0.94 to CPI of 0.97. The yearly operating expenses of the organization at that time were \$62M. Thus, the gain of three points in CPI saved about \$2M per year. Some analysts in our customer community have talked and written about this as an ROI, or return on investment, of 5 to 1, based on a \$400,000 cost for the improvement. Actually, the ROI is difficult to calculate since the savings go on year after year; on the other hand, the infrastructure expenses must also continue, but at a lower level so that the return is clearly even greater than 5 to 1. And the one-year ROI is based on reasonably hard numbers and has certainly influenced other Hughes organizations to adopt these processes. We had similar improvements for SPI.

Since that time, CPI and SPI have continued to improve. A related and very important benefit, as shown in the bottom portion of Figure 22, is that the *variation* in average SPI/CPI has continued to become smaller. This reflects less risk in our software process, more accurate bids for new programs, and increased customer satisfaction. Now, virtually all our projects run with little variability at about 1.0 for both cost and schedule. And for the exceptional case, programs that miss these marks, we look not only at program performance but the degree to which the common process is being properly followed.

A more subtle and difficult-to-measure cost saving is the degree to which the *overall program* saves money, based on the predictability of the software schedule and the absence of surprise slips and extensions. We believe that this is a large number, even larger than the direct cost savings. This cost savings, partly attributable to the reduced variability in our predictions and hence reduced risk, pleases both our internal customers—Hughes program

managers and our business leaders—and the ultimate customers to whom we deliver products containing software.

The earlier section on performance introduced the notion of review efficiency and indicated our improvements. This is important to our customers in a number of ways, including not just efficiency but risk reduction and quality improvements in the delivered product. The most important improvement for our customers was that review efficiency for requirements defects improved from 43% to 84%. At the same time, *average effort to correct the defects dropped in every phase*. The overall drop was from 0.63 staff days to 0.09 staff days. So our customers have gained two ways: first the defects are found earlier so that they are easier and cheaper to correct, and second it takes less time to correct defects no matter when they occur. In addition, our data shows that our checklists, peer reviews, and other improvements have also reduced the *absolute number* of defects in each product.

## Direct Customer Feedback

Figure 27 illustrates aspects of customer satisfaction from two important customers: one when we were just beginning to establish a strong process, and the other in the present era based on a mature and well-institutionalized process—the CSWP. The two sections below amplify on our relationships.

**Combat Grande.** We have been working hard at pleasing customers for a long time, and one early positive experience had to do not only with effective program performance, but with the customer’s recognition that we were on the leading edge of improving software development processes. The year was 1976, and the program was called Combat Grande.

Combat Grande was a highly successful Hughes program completed in 1976. It was an air defense program for Spain with over 250K SLOC applications, largely in JOVIAL. Its success was an early proof of the importance of disciplined software process based on our early process improvement. We were very concerned with metrics, even back then, for example, tracking the use of computer memory—at that time a very hot issue. An important issue, both then and now, is the tracking of trends over time to place current status in context and spot potential trouble before it becomes a serious problem. Our customer had nice things to say about us.

Hughes has been evolving a system of software management practices through their independent research and development (IR&D) program and their own project experiences. These management practices are current and well designed to provide adequate project control. In many respects, the practices are innovative and will have industry-wide implications.<sup>2</sup>

---

<sup>2</sup> “A Case Study and Evaluation of the Combat Grande Software Acquisition,” Information Systems Technology Applications Office, ESD/MCI, February 1976, p. 24.



It is significant that in 1976 we were being recognized as developing innovative practices that would beneficially influence industry.

**Peace Shield.** Hughes had a big success on Peace Shield—a tough program on a tough schedule; we had an opportunity to be awarded up to \$50 million dollars in incentives for early delivery. We also had the risk of absorbing \$50 million dollars in penalties for late delivery. Peace Shield required Hughes to (1) develop and integrate more than one million lines of operational software, (2) integrate more than 2700 major equipment units into 310 operational sites, and (3) develop a complete logistic and training support program—all in less than 54 months.

Figure 24 shows our performance on the Peace Shield contract, and illustrates the predictability of our software process based on the CPI and SPI indices. Peace Shield was planned to a 51 month bonus award schedule (rather than the 54 month contract schedule) and was managed to that date. Actual completion was 48 months—6 months ahead of contract requirements—because of continuous optimization of the software process. SPI, as well as CPI, was 0.99. Hughes received the full \$50 million dollar incentive award.

A key to our success was our mature software process, because software was the critical path from day one. Here are a few additional quotes from the Secretary of the Air Force’s letter shown in Figure 27:

*... The Air Force would like to formally announce the acceptance of Peace Shield six months ahead of the contractual obligations.*

*... and for setting a new standard for the development and deployment of large scale, software intensive systems.*

Ms. Darleen Druyun, the Acting Assistant Secretary of the Air Force for Acquisition, said that “...this is the most successful program I’ve ever been involved with, and the leadership of the U.S. Air Force agrees.”<sup>3</sup>

### **Other Aspects of Satisfying the Customer**

Our customers have shown in other ways that they are pleased with our processes, including adoption of selected processes by the Air Force as best practice, requests for process instruction at the Defense Systems Management College, and tutorials by Hughes engineers about processes and design methodologies on specific programs.

---

<sup>3</sup> *Program Manager*, DSMC Press, March-April 1996, front-page quotation of Darleen Druyun.



---

## 10. The Software Engineering Community Has Benefited From Our Investments in Software Process

Hughes Aircraft is justifiably proud of the many contributions we have made to the Software Engineering Institute, in particular, and to the software industry in general, going as far back as the 1976 *Datamation* article on the use of rate charts for software project tracking and control. Highlights of our contributions to the SEI include the most reprinted IEEE Software article, co-authored with Watts Humphrey in 1993, five consecutive full-time resident affiliates at SEI from 1989 through 1995, approximately 20 papers presented at the annual SEI conferences, video tape training modules and SEI promotions, membership in the CMM advisory board, CMM Based Assessment Board, Process Program Advisory Board, and inspiration we provided for defining the CMM level 4 and 5 key process area. We are co-founders of the Southern California Software Process Improvement Network (SPIN), co-founders and major contributors to the Irvine Research Unit for Software (IRUS), and Affiliates of the USC Center for Software Engineering.

The following pages present a comprehensive list of our contributions to the software industry.

### **Presentations and Briefings**

Alstad, J.P., "An Initial Analysis of the Personal Software Process," LA SPIN 1995, SO-CAL SPIN 1996

Bardin, B., Moon, M. F., "In search of Real Ada", International Ada Conference, 1985

Bowen, J.B., "Are Current Approaches Sufficient for Measuring Software Quality?," ACM Software Quality Assurance Workshop, 11/78

Bowen, J.B., Moon, M.F., "Experience in Testing Large Embedded Software Systems," NSIA Conference on Software Test and Evaluation, 2/83

Bowen, J.B., "AN/SPS-52B (DDG) Radar Software Reliability Study", Hughes Fullerton Final Report, FR 77-14-1106, 1/78

Bowen, J.B., Angus, J., "Software Reliability Model Study", RADC Technical Report. 1980

Bowen, J.B., Angus, J., James, L., "Combined Hardware Software Reliability Study", Hughes Fullerton RADC Technical Report 1991

Bowen, J.B., "Standard Error Classification to Support Software Reliability Assessment", NCC, 1980

Bowen, J.B., "Application of a Multi-Model Approach to Estimating Residual Software Faults and Time Between Failures", *Engineering International*, Vol 3, No. 1, 1987

Bowen, J.B., "Application of Software Reliability Models to ADCAP, Hughes Fullerton, Final report

Brown, A.W., Morris, E. J. and Rader, J. A. (1994). "The Search For Integrated CASE Environments. American Programmer," 7(7), 8-15

Camp, J.W., Jensen, E.P., "Cost of Modularity," Computer Software Engineering, Polytechnic Press, 1966

Deutsch, M.S., "An Exploratory Analysis Relating the Software Project Management Process to Project Success," IEEE Trans. on Engineering Management, Vol. 38, No 4, 11/91

Deutsch, Michael S. "Integrating the User's Concept of Operations Into a Real-Time System Definition Process," IEEE Software, September 1988, pp. 39-50.

Deutsch, Michael S. "Software Project Verification and Validation," IEEE Computer, Vol. 14(4), April, 1981, pp. 54-70.

Enos, J.C., Willis, R.R., "Tools for Embedded Software Design Verification," NASA/AIAA Workshop on Tools for Embedded Computing Systems Software , 11/78

Friedman, M. A., "Inference of and Generation from Operational Profiles for Software Reliability Testing," Proceedings 2nd Annual AIAA Aerospace Design Quality Conference, Irvine, CA, Feb. 1993.

Friedman, M. A., "Automated Software Fault-Tree Analysis of Pascal Programs," Proceedings Annual Reliability and Maintainability, Symposium, Atlanta, January 1993.

Friedman, M. A., "Applications of Hazard Rate Profile to Software Reliability Prediction, Growth Modeling, and Allocation," *Quality and Reliability Engineering International*, August 1992.

Friedman, M. A., "Automated Software Fault-Tree Analysis," Proceedings, Air Traffic Control Association 37th International Technical Program," Atlantic City, NJ, November 1992

Friedman, M. A., "Hardware/Software Reliability," Proceedings, Fourth Annual SAE International Reliability, Maintainability & Supportability Workshop," Dallas, TX, March 1992.

Friedman, M. A., "Reliability Techniques for Combined Hardware and Software Systems," RL-TR-92-15, Rome Laboratory, U.S. Department of Defense, 1992.

Friedman, M. A., "Military Handbook on Hardware/Software Reliability Assurance and Control," draft, U.S. Department of Defense, Feb. 1992.

Friedman, M. A., "Reliability Modeling of Hardware/Software Systems to Support Architectural Trade-Offs," Proceedings of the Second Annual International Symposium of the National Council on Systems Engineering, Seattle, July 1992.

Friedman, M. A., "An Overview of the Reliability Techniques for Combined Hardware/Software System Study," Proceedings of the 1992 Annual Reliability and Maintainability Symposium, Las Vegas, Nevada, Jan. 1992.

Friedman, M. A., "Modeling the Penalty Costs of Software Failure," Proceedings of the 1987 Annual Reliability and Maintainability Symposium," Philadelphia, PA, Jan. 1987.

Ginsberg, M. P., "Adopting the CMM In A Large Corporation," National SEPG Conference, Costa Mesa, Ca. 5/93

Ginsberg, M. P., "Tailoring Workshop," Software Engineering Institute, Pittsburgh, Pa. 3/94

Ginsberg, M. P., "Climbing Mount CMM," Software Technology Conference, Salt Lake City, UT. 4/95

Ginsberg, M. P., "Tailoring and the CMM," SEI Technical Report, Pittsburgh, Pa. 12/95

Ginsberg, M. P., "Tailoring In Multi-Site Organization," Three Day Workshop, Buffalo NY, 6/96

Ginsberg, M. P., "Adopting the CMM in Small Companies," Boeing Suppliers Conference, 8/96

Gotfried, Roberta, "Principle Investigations PI," DARPA Information Technology Office, January 1996.

Gotfried, Roberta, "Using Ada in Trusted Systems," COMPASS, 1993.

Hughes Aircraft Company, "Metrics - The Measure of Success," 1994

Humphrey, W.S., Snyder, T.R., Willis, R.R., "Software Process Improvement at Hughes Aircraft", *IEEE Software*, 7/91

Jensen, E.P., "The Cost of Modularity," 1975 IR&D Structured Design Methodology (Internal Hughes document)

Jensen, R., "SEER - Software Development Schedule and Cost Estimation System," 6/83

Jones, R.R., "Distributed Data Processing Modeling for Future ATC Systems," 25th ATCA, 10/81

Kacik, P. J., "An Example of Software Quality Assurance Techniques Used in a Successful Large Scale Software Development", Publication unknown, (circa 1977), referring to Combat Grande

Marcotty, M., Carpenter, M., Hersberger, R., Kelly, R., Moon, M.F., Flanders, T., "Study of Real-Time Executives for Automotive Powertrain Computer Control Applications, General Motors Research Laboratories, CS-615, July, 1990

Mauro, P., DeLeo, J., "SEPNs Improve Quality - Do They Help Productivity?," AIAA Computers in Aerospace IX Conference, 10/83

Mauro, P., "Software Quality Practices in Japan - What We Can Learn," NASA/JPL Software Quality Conference, 7/88

Moon, J.A., "Internal Process Improvement (IPI) Software Assessment, Lessons Learned", Los Angeles and Southern California SPIN 10/94

Moon, J.A., "Internal Process Improvement (IPI) Software Assessment", SEI Annual Symposium, 8/94

Moon, J.A., "Software Capability Evaluations (SCE) -- Target Organization Perspective", Los Angeles and Southern California SPIN, 10/94

Moon, J.A., "Software Risk Management" Workshop, Hughes CSI Annual Conference, 10/93

Moon, J.A., "Internal Process Improvement Software Assessment", Hughes Systems and Software Engineering Conference, 11/94

Moon, J.A., "Lessons Learned in Software Capability Evaluations -- Contractor's Perspective", SEI SCE Workshop, 10/92

Moon, J.A., "Lessons Learned in Software Capability Evaluations", SEI SEPG National Conference, 5/93

Moon, J.A., "Hughes Software Process Improvement", U.S. Navy Meeting, 11/94

Moon, J.A., "Software Capability Evaluations", Hughes CSI Annual Conference, 10/92

Moon, J.A., "Software Engineering Training", SEI Conference on Software Engineering Education, 10/92

Moon, J.A., "Software Assessment and Evaluation", Annual Software Management Conference, Litton Systems, 3/96

Moon, M. F., "Fixed-Point Made Easier", Embedded Systems Programming, Vol 8, No 3, 3/95

Moon, M. F., "Tasking Design Descriptions", Embedded Systems Programming, Vol 7, No 10, 10/94

Moon, M. F., "Buffering I/O Without Headaches", Embedded Systems Programming, Vol 6, No 7, 7/93

Moon, M. F., "Guidelines for Tasking Design", Embedded Systems Programming, Vol 6, No 10, 10/93

Naiditch, David, "Teaching Ada: Lessons Learned," presented at the Tenth Annual National Conference on Ada Technology, Atlanta, Georgia, 1992.

Noble, W.B., Leveson, N.G., "The Development of Reliable and Safe Software for Critical Applications," 1983 ATCA Convention

Patel, B.K., Tamanaha, D.Y., Rudzik, L., "Real Time System/Software Methodologies for Large Aerospace Systems", Third Annual International NCOSE Symposium, Arlington, VA, 7/93.

Patel, B.K., Holt, C., Marvel, O., "Interoperability Issues in the NATO Environment", 1989 CINC's View of Theater/Tactical C3I, Los Angeles 11/89.

Patel, B.K., Ruby, J., Tamanaha, D.Y., "Practical Experiences and Lessons Learned In Designing a Large Command Control and Information System", 1988 Aerospace Conference, Utah, 1/88.

Patel, B.K., Marvel, O., "Data Engineering in Command and Control," Proceedings for the Command and Control Research Symposium, National Defense University, Washington D.C., 6/87.

Rader, J. A. and Isaak, D. A. "An Expanded Logic Equation List for Checkout." Design Automation Conference, Miami, Florida, 1969.

Rader, J. A., "Applications of a Syntax Driver to Logic Equation Processing and Control Card Scanning." Design Automation Workshop, Washington D.C., July 17, 1968.

Rader, J. A., "Structured Design: A Case History (1974-1975)." INFOTECH, special volume on industry case studies with structured programming, 1976.

Rader, J. A., "Schematic Logic - A Software Design Notation." Eleventh Asilomar Conference on Circuits, Systems, and Computers, IEEE and Naval Postgraduate School, Pacific Grove, California, 1977.

Rader, J. A., "DRVTAX - A System for Generating Consistent Families of Language Processors." CAD 78, Third International Conference, Sussex, United Kingdom, March 14-16, 1977.

Rader, J. A., "Dynamic Allocation of Arrays in Fortran." Eleventh Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, California, 1977.

Rader, J. A., "Tools for Software Management." INFOTECH State of the Art Conference on Why Software Projects Fail, London, England, April 10-12, 1978.

Rader, J. A., "Experience with an Application of Structured Design." Proc. NCC, June 1978.

Rader, J. A., "VEVA - A Coherent Static Code Analysis System." Twelfth Annual Asilomar Conference on Circuits, Systems, and Computers, IEEE Computer Society & Naval Post Graduate School, Pacific Grove, California, November 6-8, 1978.

Rader, J. A., "Design and Documentation for Maintainability: A Case History." INFOTECH Conference on Long Life Software, Infotech International Ltd., Conference Proceedings, London, England, December 10-12, 1979.

Rader, J. A. and Pavlina, G. F., "The Contribution of Standards and Tools to Maintainability: A Case History." Asilomar Conference, November, 1979.



- Rader, J. A., "Software Tools Environments." Computer, IEEE Computer Society, 1981.
- Rader, J. A., "Management Approaches for Successfully Meeting Project Deadlines." Pergamon-Infotech State of the Art Conference, London, England, November 16-18, 1981.
- Rader, J. A., "Computer Society Workshop on the Intersection of VLSI and Software Engineering." IEEE Communications, IEEE Technical Committee on Software Engineering Newsletter, 1982.
- Rader, J. A., "DSIS - A Real-Time Radar Environment Simulation Facility," National Aerospace and Electronics Conference (NAECON), Dayton, Ohio, May, 1983.
- Rader, J. A., "VLSI and Software Engineering." Proc. Journal of Systems and Software, Elsevier Science Publishing Company, Netherlands, 1984.
- Rader, J. A., "The SDF - A Radar Avionics Software Development Environment." Proc. National Aerospace and Electronics Conference (NAECON), Dayton, Ohio, May 21-25, 1984.
- Rader, J. A., "Experiences Building and Using a Software Engineering Avionics Software." Proc. Spring Compton 85, IEEE Computer Society, San Francisco, California, February 25-28, 1985.
- Rader, J. A., "Cost Effective Software Development." Seminar on Cost Effective Software Development, Education Foundation of the Data Processing Management Association, 11-12 February Paris, 14-15 Stockholm, 18-19 London, 1985.
- Rader, J. A., "Operational Experience with ASPSS, A Simulation Facility for a Distributed Signal Processor." National Aerospace and Electronics Conference (NAECON), IEEE and others, Dayton, Ohio, May 18-22 , 1987.
- Rader, J. A., "Automatic Document Generation With Case On A DoD Avionics Project," 10th Digital Avionics Systems Conf., IEEE Computer Society Press, Los Alamitos, California, 1991
- Rader, J. A., Brown, A.W. and Morris, E. J., "An Investigation Of The State Of The Practice Of CASE Integration," Technical Report CMU/SEI-93-TR-15, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1993
- Rader, J. A., Brown, A.W. and Morris, E. J., "An Investigation Of The State Of The Practice With Respect To CASE Integration," Proc. 6th Conf. on Software Engineering Environments (SEE '93), Reading, England, IEEE Computer Society Press, 1993

Rader, J. A., Haggerty, L., "Supporting Systems Engineering With Methods And Tools," Proc. twenty-eighth Asilomar Conf. on Signals, Systems & Computers, Pacific Grove, California. IEEE Computer Society Press, 1994

Rader, J. A., Brown, A.W. and Morris, E. J., "Operational Use Of CASE Integration: An Investigation Of The State Of The Practice," J. of Systems and Software, 1995

Rader, J. A., and Brown, A.W., "Computer Aided Subprocesses (CASPs): A Practical Approach To The Use Of CASE Technology To Support Process Improvement," Proc CASE '95, Toronto, IEEE Computer Society Press

Rader, J.A., "Computer Aided SubProcesses (CASPs): Getting Process And Tools Into Operational Use," Proc Software Technology Conference, Salt Lake City, Utah, 1995

Rader, J.A., "CASE Adoption: A Process, Not An Event," in *Advances in Computers*, Academic Press, 1995

Rader, J.A., "Integration Of Process And Tools Via CASPs," NAECON'95, Dayton, Ohio, Computer Society Press, 1995

Rader, J.A., "Teaching The Elephant To Manage Requirements," Software Technology Conference, Salt Lake City, Utah, 1996

Rader, J.A., "Successful Requirements Management Collaboration," International Conference on Systems Engineering, 1996

Rader, J.A. and Fowler, P., "How Tools Can Enhance The Requirements Management Process: CASPs Transition Packages And The Process Change Guide," Software Engineering Institute Software Symposium, Pittsburgh, PA, 1996

Rova, R.M., "Utilizing Two-Person Teams," Navy's Best Manufacturing Practices Workshop, 1990

Rova, R.M., "Process Improvement using Defect Data," Tuscon, AZ, NCOSE meeting, 1995

Rova, R.M., "Overview of Quantitative Process Management at Hughes Fullerton", SEI Software Symposium, Pittsburgh, PA, 1994

Shumate, K., Snyder, T.R., "Software Project Reporting: Management, Measurement, and Process Improvement," Tri-Ada '94, 11/94.

Snyder, T.R., "Rate Charting," *Datamation*, 11/76

Snyder, T.R., Shumate, K., "Kaizen Project Management," *American Programmer*, 12/92

Stickney, M.E., "An Application of Graph Theory to Software Test Data Selection",  
publication unknown, 1978

Tamanaha, D.Y., "Analyzing Systems for Understandability: Structured Process Flows",  
Third Annual International NCOSE Symposium, Arlington, VA, 7/93. NCOSE "Outstanding  
Paper Presentation, Automation And Tools" Award

Tamanaha, D.Y., "Structured Process Flows (SPFs): A Process Model for Metrics", 1991  
IEEE Aerospace Conference Digest, Crested Butte, CO, 2/91.

Tamanaha, D.Y., Patel, B.K., Litchfield, V., Davis, A., "System Methodologies for Large  
Aerospace Systems", 1991 IEEE Aerospace Conference Digest, Crested Butte, CO, 2/91.

Tamanaha, D.Y., "A Platform for Software Metrics: Structured Process Flows (SPFs) of  
Software Process Models", Proceedings of the Structured Development Forum XI, San  
Diego, CA, 4/90

Tamanaha, D.Y., Bourgeois, P., "Rapid Prototyping of Large Command, Control,  
Communications and Intelligence C3I Systems", 1990 IEEE Aerospace Conference Digest,  
Vail, CO, 2/90

Tamanaha, D.Y., Patel, B.K., Dana, J.L., "A System Data Engineering Methodology (SDEM)  
for Command and Control Systems", Proceedings of the 1989 Symposium on Command and  
Control Research, Wash. D.C., 7/89.

Tamanaha, D.Y., Patel, B.K., Wenjen, W. "The Application of CASE to Large Aerospace  
Systems", IEEE Aerospace Conference Digest, Breckenridge, CO, 2/89.

Tamanaha, D.Y., "An Integrated Rapid Prototyping Methodology for Command and Control  
Systems: Experience and Insight", Proceedings of IEEE HICSS-22, 1/89

Tamanaha, D.Y., Lew, A., "A Note on Proving the Correctness of a Decision Table Program,"  
Proceedings of the 15th Asilomar Conference, 11/81

Tamanaha, D.Y., Lew, A., "Decision Table Programming and Reliability", Proceedings of  
2nd Int. Conf. on Software Engineering, 10/76

VanTilburg, R.L., Winchester, J.W., "On Evolution of New Technology and Problems,"  
Symposium on System Development Methodology, 4/77

Willis, R.R., "DAS - An Automated System to Support Design Analysis," 12th Asilomar Conference on Circuits, Systems, and Computers, 11/78

Willis, R.R., Jensen, E.P., "AIDES - Computer-Aided Design of Software Systems," *Software Engineering Environments*, North-Holland, 1981

Willis, R.R., "Technology Transfer Takes  $6\pm 2$  Years," IEEE Software Technology Transfer Workshop, 4/83

Willis, R.R., "Reducing the Risks for Software Quality on FAA's AAS," NSIA Conference, 10/87

Willis, R.R., "Case History and Lessons Learned in Process Maturity Improvement," NSIA Conf. on Software Quality and Productivity, 4/90

Willis, R.R., "Software Process Maturity - Our First Step Toward Solving the Software Problem," 1st Annual Software Technology Conference, 1990

Willis, R.R., "Experience in SEI Process Maturity Assessment," SEI Affiliate's Symposium, 9/90

Willis, R.R., "The Road to Level 3 and Up," AIAA Total Quality Management Symposium, 11/90

Willis, R.R., "Standards Help Engineers, Standards Hinder Artists," NSIA Conference on Software Quality and Productivity, 4/91

Willis, R.R., "Quantitative Process Management - Lessons Learned," 3rd Rome Labs Quality Workshop, 8/91

Willis, R.R., "Lessons Learned in Software Process Improvement," Conference on Strategic Software Systems, 3/92

Willis, R.R., "Software Metrics in SED," FSSEC Process Improvement Colloquium, Fort Sill, 5/92

Willis, R.R., "Lessons Learned at Level 4/5 Process Maturity," SEI Software Engineering Conference, 10/95

Woodall, Thomas R. and Gotfried, Roberta, "The Privilege Control Table Toolkit: An Implementation of the System Build Approach," Nineteenth National Information Systems Security Conference, Baltimore, Maryland, 1996.

Wright, V.E., "Training Programs and Plans for Software Process Improvement", SEPG National Conference, 4/93

Wright, V.E., "Training Programs and Plans for Software Process Improvement", So. Cal. SPIN, 6/93

Yin, B.H., Winchester, J.W., "The Establishment and Use of Measures to Evaluate the Quality of Software Designs," ACM: SIGMETRICS and SIGSOFT Conference on Software Quality, 11/78

## **Books**

Deutsch, M.S., *Software Verification and Validation*, Prentice Hall, 1984

Deutsch, M.S., Willis, R.R., *Software Quality Engineering*, Prentice Hall, 1988

Friedman M. A., Voas, J., *Software Assessment: Reliability, Safety, Testability*, John Wiley & Sons, 1995

Friedman M.A., Tran, P., Goddard, P., *Reliability of Software Intensive Systems*, Noyes Press, 1995

Jensen, R, Tonies, C., *Software Engineering*, Prentice Hall, 1980

Krell, *Developing with Ada*, Bantam, 1992.

Naiditch, D., *Rendezvous with Ada 95*, John Wiley & Sons, 1995.

Naiditch, D., *Rendezvous with Ada: A Programmer's Introduction*, John Wiley & Sons, 1989.

Nielsen, K., *Object-Oriented Design with Ada*, Bantam, 1992.

Nielsen, K., *Ada in Distributed Real-Time Systems*, McGraw-Hill, 1990.

Shumate, K., *Understanding Ada*, Harper & Row, 1984.

Shumate, K., *Understanding Concurrency in Ada*, McGraw-Hill, 1988.

Shumate, K., Nielsen, K., *Designing Large Real-Time Systems with Ada*, McGraw-Hill, 1988.

Shumate, K., *Understanding Ada: with Abstract Data Types*, 2nd Edition, John Wiley & Sons, 1989.

Shumate, K., *Software Specification and Design*, John Wiley & Sons, 1992.

## **Partnerships**

Software Engineering Institute Strategic Partner

Software Productivity Consortium (SPC) Member

Microelectronics Computer Consortium (MCC) Associate

Irvine Research Unit for Software (IRUS) Member

USC Center for Software Engineering Affiliate

SEI, Cooperative Research and Development Agreement (CRADA) for requirements management process supported by DOORS

Hughes Information Technology Systems funds research at the University of Maryland's Department of Computer Science in software reuse and interoperability design patterns; Dr. Victor Basili and Dr. Gianluigi Caldiera are principal investigators

## **Professional Associations**

Ada 9X Distinguished Reviewers, Bardin, B.

Ada Interest Group, Bardin, B.

Ada Rapporteur Group (ARG), Bardin, B.

AIA Embedded Computer Software Committee (ECSC), T. Snyder, Chairman

AIAA Software Systems Technical Committee, G. Barksdale, Emeritus

ANSI Ada TAG, B. Bardin

Business Management Council, C. Flora

Cal State Fullerton (CSUF) Corporate Key Executive, T. Snyder

Editorial Board of Journal of Systems and Software, Rader, J.A.

EIA Computer Resources Committee, R. Fanning

EIA Industry Reuse Advisory Group (IRAG), K. Shumate

IEEE standard for productivity metrics (1984-86), P. Stevens, R. Willis

International Standards Organization (ISO) SW Engineering Standards (SC7) U.S. TAG, R. Willis

International Standards Organization (ISO) "SPICE" project (1994), R. Willis

Irvine Research Unit and Software Executive Leadership Committee, T. R. Snyder - President

Irvine Research Unit in Software (IRUS), R. Willis

National Institute of Standards & Technology, Integrated Software Engineering Environments Working Group, Rader, J.A., 1993-95

NCOSE Tools Integration Working Group, Rader, J.A., 1995-

NSIA, Software & Information Systems Executive Council, G. Barksdale

OMG Runtime Performance Working Group, Finn, D., chairperson

POSIX Language Bindings Working Group, Greene, R., Vice Chairperson

Quantitative Process Management, Willis, R.R., videotaped guest lecture at SPC, 1993

SEI CMM Advisory Board (CAB) inaugural membership (1988-1991), R. Willis

SEI CMM Based Assessment (CBA) Advisory Board, Moon, J.A.

SEI Process Program Advisory Board membership, Snyder, T.R.

SEI Resident Affiliates, M.S. Deutsch, R.W. Lichota, J.A. Rader, J.P. Alstad, M.P. Ginsberg,

SEI SCE Advisory Board, Moon, J.A.

Southern California Software Process Improvement Network (SPIN), J. Youmans

Teamwork Users Group Executive Board, Rader, J.A., 1989-94

Technical Advisory Board of the Software Productivity Consortium, P. Stevens

UCI Executive Committee, Snyder, T.R.

UCI Information & Computer Science (ICS) American Indian Summer Institute in Computer Science (AISICS) Corporate Steering Committee, S. Acterman

## Contributions

Accreditor for the Computer Sciences Advisory Board, Alstad, J.P.,

Allied Signal Canada (ASACa) Contract for Process Improvement, Willis, R.R., L. Frassetto, 4/93 - 7/94

Applying Total Quality Management to Software Systems, Deutsch, M.S. Tutorial: The 13th, 14th, and 15th International Conferences on Software Engineering; April 1991, Austin, Texas; May 1992, Melbourne, Australia; May 1993 Baltimore, Maryland

Chairperson of the Society of Automotive Engineers OS API Working Group (SAE AS5-2B), Schleicher, D.

Defense Systems Management College, Training for "Software Risk Management", Moon, J.A., 91-95

DOORS Wolfpack (Advisory Board), Rader, J.A., 1996-

Extending Into the Software Systems Business, Deutsch, M.S. Motorola Corp.'s Software Management Seminar, July 1995 and March 1996, Phoenix, Arizona

Future of Airborne Information Security, Gotfried, R., Chair, National Aerospace and Electronics Conference (NAECON), May, 1996.

Future OS Working Group sponsored by NSA, DARPA, and DISA, Gotfried, R., member

IEEE *Software* reviewer, R. Willis

Irvine Reassert Unit for Software (IRUS) sponsorship and leadership, Hughes Software Program Council

Software Metrics, Willis, R.R., Lobitz, B.O., Rova, R.M., 3-day course, San Diego State University, 1994

Software process for Opel, Germany, Joseph, A. Helping to define for GM

Software Process Maturity Improvement, Willis, R.R., a 3-day seminar by S.P. Seminars, 1992 given to 120 engineers



Software Program Managers Network booklet, "The Program Manager's Guide to Software Acquisition Best Practices", Snyder, T.R., Fanning, R.F., Rova, R.M., Cooperman, R., Shumate, K.

Software Project Management: The Empirical Success Factors, Deutsch, M.S. Tutorial: COMPEURO 90 - 1990 IEEE International Conference on Computer Systems and Software Engineering, May 1990, Tel-Aviv, Israel.

Software Quality Assurance, Levine, L.P., 3-day course, San Diego State University

Some Political, Sociological, and Technical Aspects of Testing, Deutsch, M.S. Keynote address: Fifth International Conference on Testing Computer Software, June 1988, Washington D.C.

Southern California SPIN sponsorship and leadership, Hughes Software Program Council

Tri-services Joint Integrated Avionics Working Group (JIAWG) Software Engineering Environments Team, Rader, J.A., 1990-93

Use of Metrics in Software, Willis, R.R., Keynote speaker, Motorola software conference, 1993

## **Benchmarks**

Texas Instrument, Defense Software Engineering Group, J.D. Grimm and D.J. Frailey (TI) and H.Griswold (Hughes) coordinators, 10/93

Hewlett-Packard, Corporate Software Engineering Group, S. Stetak (HP) and R. Willis (Hughes) coordinators, 2/91

NEC

Allied Signal, Canada, Software Engineering Department, L. Frassetto (ASACa) and R.R. Willis (Hughes) coordinators, 4/93

AT&T, Bell Research Lab and International Switching, Dewayne Perry (AT&T) and Ben Lobitz (Hughes) coordinators, 8/93

## **Recognitions**

Air Force selection as "Best Practice" in software

Recognized by DoD Software Acquisition "Best Practice" for quantitative process management based on Project Reporting procedure, 9/94

California ETP grant for software training, 1995

Invited as speakers at Motorola, SPC, National SEPG, SEI symposia 1990-1996

Recognition by Darleen Druyen for Peace Shield success

Recognition by George Donahue for WAAS success

Recognition by Lloyd Mosemann for PRISM success

## **Innovations**

Hughes Aircraft has been responsible for many innovations in the field of Software Process Improvement, including:

**Rate Charts** – The software rate chart is considered the most important software planning and status tool ever used at Hughes Aircraft. Its application to the software development was invented by Don Ormond of the Computer Programming Laboratory (CPL) in 1972, as part of his master’s thesis. It was refined and applied by Terry Snyder on the successful Combat Grande program and first published in *Datamation* in 1976. Several versions of rate chart automation are also considered innovations. The latest, The Management Information Report Generator (TMIRG), was used on the successful Peace Shield program.

**Quantitative Process Management** - The phrase “Quantitative Process Management” (QPM) was first coined in Hughes Software Engineering Division following the 1987 assessment. Sally Cheung and, later, Bob Lanphar were assigned to make the concept of QPM a reality, as part of the action plan resulting from the assessment. This innovation, and the concurrent collaboration by Ron Willis on the definition of the CMM level 4 KPAs (then called Measurement and Analysis of Software) in 1988-1989, led to the definition of the current CMM capability: QPM.

**Software Quality Engineering** – On the FAA Advanced Automation System (AAS) contract conducted in the Software Engineering Division (SED) during 1984-1985, introduced, documented, and partially implemented the idea of setting goals (actually requirements) for customer perceived software quality attributes, and managing the software process to achieve those goals. That process was captured in the book, *Software Quality Engineering* (Deutsch, Willis). These concepts influenced the definition of the CMM level 4 capability, Software Quality Management.

**Project Review** - The software project review was conceived, piloted, formalized, and institutionalized by the SED in the 1970s. Since then it has been used and continuously improved as the most effective way to monitor and control software development. Those who understand its importance see the project review as the root of process maturity in our

organization. The tool, QPMIS, developed to help automate the approximately 20 reports of the project review, is another innovation.

**First Organization to Level 3** – Our SED was the first organization to achieve a level 3 process maturity rating. The SEI-assisted assessment resulting in the level 3 rating was led by Watts Humphrey in January 1990; the findings are documented in SEI/CMU-90-SR-5.

**Object Oriented Competencies** - Hughes is applying object-oriented methods to the Earth Observing Data and Information Core System (ECS), a NASA sponsored system that is acquiring the multi-disciplinary database supporting global climate change research. ECS is one of the largest applications of object oriented (OO) techniques in the world to-date, consisting of 1.2 million source lines of custom-developed code integrated with more than 60 commercial-off-the-shelf applications supporting an eventual archive of 1.4 petabytes. Several key process competencies have been derived from this work:

- extension of the basic Rumbaugh methodology to a major "composite system" process entailing integration of commercial-off-the-shelf packages, generated code, legacy system components, glue code, and custom-developed code
- trained more than 400 software engineers, system engineers, and customer personnel in OO analysis, design, and C++ software development
- devised an organization and technical process that identified and implemented common object classes reducing code development by 25% in the first release
- applied a reuse process that was able to transfer major ECS components to a system sponsored by another customer who had similar requirements; this was partially enabled by the use of OO techniques

### **Additional Results of Our Process Improvements**

For 25 years Hughes Aircraft has focused on improving our software development process. During this period, our influence has affected other Hughes and industry organizations in a positive way:

**Reengineering Influence on the Hughes Engineering Community** – The capabilities, lessons learned, formal process descriptions, tools, and skills in software process maturity have been the cornerstone of Hughes Aircraft’s process reengineering efforts for all engineering disciplines. Thirteen engineering disciplines, from mechanical engineering to optics and lasers, have followed software’s lead in developing process management for their discipline. Key among these capabilities are: process-centered engineering, tailoring common processes to customer requirements, process management, and continuous measurable improvement.

**CSWP as a Model for System Engineering Process** - To achieve a level 3 system engineering capability as soon as possible, Hughes adopted the formal software process description as their starting point to “reengineer” systems engineering. Approximately 100

software practices and procedures were, literally, copied and edited, to become system engineering's first-cut directive system.

**Influence on the CMM** – Many of Hughes recognized software experts have contributed to the design and intellectual content of the CMM. A few examples include: Ron Willis, who contributed to the questionnaire and level 4 and 5 capabilities; Jane Moon, who contributed to the CBA IPI process; and Mark Ginsberg, who contributed to CMM tailoring for small projects.

**Helping to Define “Six Sigma” for Software** – Our long-time interest in collecting, reporting, and analyzing software defect data prepared us for the “Six Sigma” revolution started by Motorola. The application of Six Sigma principles to software was pioneered by leading metrics experts at Hughes, including Robert Rova and Rich Fanning. The software Six Sigma approach pioneered by Hughes has influenced many of the industry's leading Six Sigma advocates, including Motorola and Texas Instruments. Software Six Sigma is taught as a standard module in our corporate education curriculum and the tool, QIP, is used to help automate the six sigma data collection and reporting.

**Leading the Way for Quantitative Process Management** – Our long-time interest in software measurement and analysis helped us to define the level 4 and 5 CMM capabilities and serve as proof that quantitative techniques applied to software result in controlled, continuously improving software processes.

---

## 11. Abbreviations

<b>AAS</b>	Advanced Automation System: upgrade to FAA's National Air Management System
<b>ACM</b>	Association for Computing Machinery: organization of computer professionals
<b>ADCAP</b>	Advanced Capability Torpedo: Hughes contract to build modern torpedo for U.S. Navy
<b>AIA</b>	Aerospace Industries Association
<b>AIAA</b>	American Institute of Aeronautics and Astronautics
<b>AIDES</b>	Automated Interactive Design and Evaluation System: CASE tool for SW design conceived, developed, and implemented by Hughes Software Engineering Division in early 1980's
<b>ISIS</b>	American Indian Summer Institute in Computer Science: part of UCI
<b>ANSI</b>	American National Standards Institute
<b>ASAC</b>	Allied Signal, Canada: part of Allied Signal company
<b>AT&amp;T</b>	American Telephone and Telegraph
<b>ATC</b>	Air Traffic Control
<b>ATCA</b>	Air Traffic Control Association
<b>CAB</b>	CMM Advisory Board: industry board to help guide CMM development
<b>CASE</b>	Computer-Aided Software Engineering
<b>CASP</b>	Computer-Aided Subprocesses: method for process capture
<b>CBA</b>	CMM-Based Assessment

<b>CD</b>	Compact Disk
<b>cmi</b>	continuous measurable improvement: Hughes guiding principle
<b>CMM</b>	Capability Maturity Model: collaboratively developed framework of software best practices
<b>CMU</b>	Carnegie Mellon University: parent unit of SEI
<b>CPI</b>	cost performance index: measurement unit of development progress
<b>CPL</b>	Computer Programming Laboratory: 1970s name of SED
<b>CRADA</b>	Cooperative Research and Development Agreement: SEI-industry collaboration
<b>CSI</b>	Corporate Software Initiatives: early name of SSEC
<b>CSUF</b>	California State University at Fullerton
<b>CSWP</b>	common software process: Hughes common engineering process for software
<b>DARPA</b>	Defense Advanced Research Projects Agency
<b>DAS</b>	Design Analysis System: Hughes CASE tool for analysis of system designs
<b>DIS</b>	Distributed Information System: part of ECS
<b>DISA</b>	Defense Industrial Security Agency
<b>DoD</b>	Department of Defense
<b>DOORS</b>	Dynamic Object Oriented Requirements System
<b>DSMC</b>	Defense Systems Management College
<b>ECS</b>	Earth Observing Data and Information Core System: Hughes contract with NASA for earth sciences information system gathering and world-wide distribution
<b>EIA</b>	Electronic Industries Association
<b>ESD</b>	Electronic Systems Division: part of the Air Force

<b>ETP</b>	Employee Transition Program: state-funded employee training program
<b>FAA</b>	Federal Aviation Administration: managers of air traffic in U.S.
<b>FSSEC</b>	Field Systems and Software Engineering Center: part of Ft. Sill organization.
<b>GM</b>	General Motors: automobile manufacturer with 850,000 employees
<b>HP</b>	Hewlett Packard: company making electronic devices
<b>IEEE</b>	Institute of Electrical and Electronic Engineers: professional organization
<b>IPI</b>	Internal Process Improvement: company's self-initiated software process improvement
<b>IR&amp;D</b>	Independent Research and Development: profit invested in product improvement
<b>IRAG</b>	Industry Reuse Advisory Group: part of EIA
<b>IRUS</b>	Irvine Research Unit for Software: organization to foster industry-university collaboration
<b>ISO</b>	International Organization for Standardization: the ultimate rules maker
<b>JIAWG</b>	Joint Integrated Avionics Working Group: DoD team for common aircraft avionics
<b>JPL</b>	Jet Propulsion Laboratory: an FFRDC for NASA
<b>KPA</b>	key process area: a part of the CMM
<b>LA-SPIN</b>	Los Angeles area SPIN: one of two SPINs in So. California associated with USC
<b>MPRP</b>	Monthly Program Review Package: 20 reports from Project Review
<b>NAECON</b>	National Aerospace and Electronics Conference
<b>NASA</b>	National Aeronautics and Space Administration

<b>NATO</b>	North Atlantic Treaty Organization: post-World War II entity to manage peace
<b>NCC</b>	National Computer Conference: one of first, largest computer professional conferences
<b>NCOSE</b>	National Council on Systems Engineering
<b>NEC</b>	Nippon Electronics Company
<b>NSA</b>	National Security Agency: part of the U.S. government
<b>NSIA</b>	National Security Industry Association
<b>OO</b>	Object Oriented: current software design principle
<b>PDT</b>	Process Deployment Team
<b>POC</b>	Process Owner Council
<b>QIP</b>	Quality Indicator Program: Hughes software defect information system
<b>QPMIS</b>	Quantitative Process Management Information System: Hughes QPM system
<b>RADC</b>	Rome Air Development Center: FFRDC for the Air Force
<b>ROI</b>	return on investment: how long it takes before an investment pays for itself; ratio of payoff to the investment required to produce the payoff
<b>ROM</b>	rough order of magnitude: educated guess
<b>STAC</b>	Software Technology Advisory Council
<b>SWPT</b>	Software Engineer Process Team
<b>SAE</b>	Society of Automotive Engineers
<b>SCE</b>	Software Capability Evaluation: SEI-defined contractor's audit of supplier SW capability
<b>SDCE</b>	Software Development Capability Evaluation: an ASC replacement for SCE



<b>SDEM</b>	system data engineering methodology: Hughes term for data engineering
<b>SDF</b>	Software Development Facility: old terminology found in DoD-STD-2167
<b>SED</b>	Software Engineering Division: Hughes organization. where CSWP was invented and proven
<b>SEER</b>	software estimation and evaluation of resources
<b>SEI</b>	Software Engineering Institute: FFRDC for software technology transfer
<b>SEPG</b>	Software Engineering Process Group: team focused on software process
<b>SEPN</b>	Software Engineering Procedures Notebook: earlier form of SEPP
<b>SEPP</b>	Software Engineering Practices and Procedures: earlier form of CSWP
<b>SIGMETRICS</b>	Special Interest Group in Software Metrics: part of ACM
<b>SIGSOFT</b>	Special Interest Group in Software: part of ACM
<b>SLOC</b>	source line of code: unit of measure for software size
<b>SOCAL-SPIN</b>	Southern California SPIN: one of two SPINs in So. California associated with UCI
<b>SPC</b>	Software Productivity Consortium: industry consortium to fund technology improvements
<b>SPF</b>	structured process flows: process flow diagramming
<b>SPI</b>	software process improvement: term used by SEI
<b>SPICE</b>	Software Process Improvement and Capability dEtermination: ISO's "CMM"
<b>SPIN</b>	Software Process Improvement Network: consortium of local SW process advocates
<b>SPM</b>	software project manager: leader of software development on a

	project
<b>SQA</b>	software quality assurance: function performed by people who verify SW process and product compliance
<b>SSEC</b>	Systems and Software Engineering Council: Hughes corporate-level SEPG
<b>SW</b>	software
<b>TI</b>	Texas Instruments
<b>TMIRG</b>	The Management Information Report Generator: current version Hughes MIRG
<b>ToF</b>	team of four: team approach for process deployment
<b>TQM</b>	Total Quality Management: 1980s quality initiative
<b>UCI</b>	University of California at Irvine
<b>VLSI</b>	very large scale integration
<b>WAA</b>	wide area augmentation
<b>WAAS</b>	Wide Area Augmentation System: FAA's use of GPS for air traffic control

---

# Bibliography

**[ESD/MCI 76]**      “A Case Study and Evaluation of the Combat Grande Software Acquisition,” Information Systems Technology Applications Office, ESD/MCI, February 1976.

**[DSMC 96]**      *Program Manager*, DSMC Press, March-April 1996.



# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (LEAVE BLANK)		2. REPORT DATE May 1998	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process		5. FUNDING NUMBERS C — F19628-95-C-0003	
6. AUTHOR(S) R.R. Willis, R. M. Rova, M.D. Scott, M.I. Johnson, J.F. Ryskowski, J.A. Moon, K.C. Shumate, T.O. Winfield			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-98-TR-006	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/DIB 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER ESC-TR-98-006	
11. SUPPLEMENTARY NOTES			
12.A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12.B DISTRIBUTION CODE	
13. ABSTRACT (MAXIMUM 200 WORDS) This report describes the software improvement activities of Hughes Aircraft Company <sup>1</sup> over the last 25 years. The focus is on continuous improvement of the software development process and the deployment of that process from a single organization at Fullerton, California, to virtually all the 5000 software engineers of Hughes Aircraft. For this achievement, the widespread deployment of a continuously improving software process, Hughes Aircraft was awarded the 1997 IEEE Computer Society Software Process Achievement Award.			
14. SUBJECT TERMS software improvement, software development, software process improvement, continuous improvement, project management, program review, performance improvement, customer satisfaction		15. NUMBER OF PAGES 87	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED
		16. PRICE CODE	
20. LIMITATION OF ABSTRACT UL			

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18  
298-102

<sup>1</sup> At the time of the submittal of this paper, the authors were with Hughes Aircraft Company, which has since merged with Raytheon.

