When Management Gets Serious About Managing Software

P. A. "Trisha" Jansma
Software Quality Improvement Project
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109-8099
818-354-0647
Patti.A.Jansma@jpl.nasa.gov

Abstract—In FY 2002, JPL launched a lab-wide software quality improvement effort aimed at addressing the challenges of developing, managing and acquiring software in an engineering and scientific environment. The lab's senior management formed the Software Quality Improvement (SQI) Project in order to establish an ongoing operational effort that results in continuous, measurable improvement of software quality at JPL. After some study, the SQI Project identified several factors that are critical for the successful and timely management and implementation of software-intensive systems.

The initial focus of their efforts was the basic project management of software classified as mission-critical or mission support. The areas of emphasis selected were based on the CMMI® model, and included software project planning, software project monitor and control, software acquisition management, software risk management, software requirements management, software quality assurance, software configuration management, and software measurement and analysis. The SQI Project developed a broad range of products, services, and training to support managers and practitioners throughout the entire software development life-cycle.

As products, services and training were developed, each one needed to be systematically deployed. Hence, the SQI Project developed a deployment process that includes four aspects: infrastructure and operations, communication and outreach, education and training, and consulting support. In addition, the SQI Project took a proactive approach to organizational change management and customer relationship management – both concepts and approaches not traditionally invoked in an engineering environment. This paper [1, 2] describes JPL's approach to improving how software is managed, developed and acquired. It discusses the various products, services and training that were developed, describes the deployment approach used, and concludes with several "lessons learned".

[1] 0-7803-8870-4/05/$20.00© 2005 IEEE

[2] IEEEAC paper #1025, Version 11, December 10, 2004

TABLE OF CONTENTS

1. INTRODUCTION 1
2. TARGET AUDIENCE 2
3. GENERAL APPROACH 3
4. DEPLOYMENT PROCESS 8
5. INFRASTRUCTURE AND OPERATIONS 8
6. COMMUNICATION AND OUTREACH 10
7. EDUCATION AND TRAINING 12
8. PROJECT SUPPORT 13
9. LESSONS LEARNED 14
10. CONCLUSIONS 15
11. ACKNOWLEDGEMENTS 15
12. ACRONYMS AND ABBREVIATIONS 16
REFERENCES 16
BIOGRAPHY 17

1. INTRODUCTION

About JPL

The Jet Propulsion Laboratory (JPL), located in Pasadena, California, is a non-profit federally funded research and development center which is operated by the California Institute of Technology under a contract with the National Aeronautics and Space Administration (NASA). JPL is part of the U.S. aerospace industry, and is NASA's lead center for robotic exploration of the solar system. In addition to its work for NASA, JPL performs tasks for a variety of other federal agencies, such as the U.S. Department of Defense (DOD), the Department of Transportation (DOT), the Department of Energy (DOE), etc. JPL has approximately 5500 employees: 4500 in the technical and programmatic divisions and 1000 in the administrative divisions. Its annual budget is approximately $1.4 billion.

Background

Motivated by some highly visible failures in which software was implicated in mission loss (e.g., Mars '98) and by a NASA-wide software engineering initiative, JPL undertook a major software quality improvement effort. JPL senior management formed the Software Quality Improvement (SQI) Project in FY 2002 in order to establish an on-going operational effort that results in continuous, measurable

improvement of software quality at JPL. The SQI Project is chartered to provide education, training, mentoring, and consulting for projects and practitioners in order to enable and promote software best practices, and to leverage JPL experience in software engineering in support of major software projects, throughout the entire software life-cycle.

The SQI Project itself consists of the following components:

1. SQI Project Management that manages the SQI Project and all its activities, and communicates with JPL senior management and with other external interfaces.
2. Process and Product Definition Element that captures, defines, and refines repeatable processes and a set of engineering and management practices for project use.
3. Measurement, Estimation and Analysis Element that provides infrastructure for software estimation, costing and measurement; and collects and analyzes measures of development performance.
4. Deployment Element that promotes communication and infuses practices into project use; provides education, training and consulting for projects; and provides SQI Project infrastructure.
5. CMMI® Implementation Team that implements various CMMI® practices and conducts appraisals against the CMMI® model.

The SQI Project is very fortunate to have the four "must haves" necessary for success in creating major change defined by Hutton [9]:

1. A compelling reason for change – NASA agency-wide initiative, Caltech JPL Advisory Group recommendations, recent highly visible failures and corresponding accident reports
2. Suitable sponsors – Associate Director for Flight Projects and Mission Success, and Directors for program and technical Directorates
3. Informed commitment of sponsors – substantial burden funding and active commitment of Directors and senior management
4. A change agent or "champion" – Process Owner for the Develop Software Products (DSP) process.

Also, they were able to build on some previous process improvement activities at JPL in the 1980's and 1990's, including Total Quality Management (TQM), Process-Based Management (PBM), ISO 9000 certification, and the Software Resource Center (SORCE). However, they had to deal with some "baggage" associated with these previous initiatives as well.

2. TARGET AUDIENCE

JPL's employees are classified into 13 job families, and each family has several disciplines and sub-disciplines. While the majority of the JPL Software Community consists of practitioners in the Information Systems and Computer Science (IS&CS) job family, software managers are categorized as either Line Management or Program/Project Management. Also, personnel who are categorized as Engineering and Technical would still be considered part of the Software Community provided that at least 50% of their work is software-intensive. Given this range of categories, the software community at JPL consists of approximately 1700 people. However, the SQI Project narrowed their concept of their target audience within the software community to exclude three categories, namely those who are:

- developing software for research and development (R&D) purposes,
- involved in information technology (IT), e.g., computers, networks, databases, and security, and
- providing customer support e.g., database administration, system administration, user support, and training.

This exclusion involves almost 200 people, bringing the target audience size down to approximately 1500 people.

Use of Customer Relationship Management (CRM)

The SQI Project took a very proactive approach to customer relationship management, an approach not traditionally invoked in an engineering environment. For those not familiar with this concept, customer relationship management (CRM) is "a strategy used to learn more about customer's needs and behaviors in order to develop stronger relationships with them." [5] It helps ensure that all products and services truly provide value to the customer, and that the "real" customers are being targeted and reached. CRM defines a customer acquisition life-cycle with six stages labeled unaware, awareness, interest, action, user, and evangelist. CRM involves identifying and prioritizing customer target segments, creating customer profiles, and tailoring the communication messages and approach to their specific needs [4], [7], [16].

The SQI Project identified five customer segments as shown in Table 1. Software management customers include Project Element Managers (PEMs), Project Software Systems Engineers (PSSEs), software managers, mission assurance managers (MAMs) and Level I line managers (supervisors) of software-intensive organizations. Software engineering customers include Cognizant Engineers (Cog Es), software architects, software engineers, software test engineers, software configuration management (CM) engineers, and software quality assurance (SQA) engineers. Project management customers include managers in JPL program and project offices whose purview is broader than software, but whose scope encompasses it as well. Usually these managers have a hardware background and would benefit from exposure to the fundamental concepts associated with software management and planning. Hence, other customers include program managers, project managers, systems engineers, others with whom software

Table 1 Customer Segments in the SQI Target Audience

#	Customer Segment Name	Target Audience	Job Family/ Career Level
1	Mid-level Line Management Customers	• Section Managers of software-intensive sections. • Deputy Section Managers of same	Line Management / Manager II
2	Project Management Customers	• Project Managers of projects with major software components.	Project Mgmt. / Manager II or III
3	Software Management Customers	• Project Element Managers (PEMs) • Project Software Systems Engineers (PSSEs) • DSMS Task Managers • DSMS Operations & Engineering Managers (formerly Service System Manager) • Software Systems Engineers • Software Managers • Mission Assurance Managers (MAMs) • DSMS System Engineers (SE) • DSMS Subsystem Engineers (SSE) (service or element) • Technical Group Supervisors (TGSs) of software-intensive groups. *Note some Cog Es and Contract Technical Managers (CTMs) may fall into this category.*	Project Mgmt. / Manager I, or Line Management / Manager I, or IS&CS, Technical, or Engineering / Principal, Senior A, or Senior
4	Software Engineering Customers	• Cognizant Engineers (Cog Es) • DSMS Cognizant Development Engineers (CDEs) • Software Architects • Software Engineers • Software Test Engineers • Software Configuration Management (CM) Engineers • Software Quality Assurance (SQA) Engineers	IS&CS, Technical, or Engineering / Senior, Staff, or Associate
5	Process Improvement Customers	• SQI Project personnel • Software Process Engineers • CMMI Appraisal Team members • Systems Engineers • SQA Engineers	varies

personnel interact regularly, and *anyone whose decisions impact the way software is developed at or acquired by JPL.* Lastly, it includes selected members of the Acquisition Division involved with acquiring software or systems with embedded software.

The SQI Project prioritized their outreach based on classes of software and application domains as shown in Table 2. Their primary customers are those working on mission-critical software for spacecraft, instruments, and associated ground systems. Their secondary customers are those working on mission support software, and their tertiary customers are those working on enterprise, business or administrative software. Using these definitions, they characterized all sections within the Engineering and Science Directorate, Mission Assurance Office, and Institutional Business Systems Division as either software-intensive (S), partial software (P), or no or very limited software (N), and then identified whether they were part of their primary (1), secondary (2) or tertiary (3) audience. After they had characterized their target audience and established their priorities, they tailored their training and presentations to the unique needs of each segment, and worked to focus their initial efforts on their primary audience.

3. General Approach

The SQI Project identified several factors that are critical for the successful and timely management and implementation of software intensive systems. The initial focus of their efforts is the basic project management of mission-critical software based on the CMMI® model. The

Table 2 SQI Priorities vs. Class and Type of Software

Priority	Software Class	Sample Types of Applications
1	Mission Critical Software	Flight software (guidance and control, command and data handling, fault protection, payload interface, instrument control, on-board data processing, real-time operating systems) and tracking and navigation software, etc.
2	Mission Support Software	Science instrument, science data processing, science analysis, simulation, and telecommunications software, etc.
3	Enterprise, Business or Administrative Software	Project Data Management System (PDMS), Electronic Libraries, Financial, Acquisition and Human Resources applications, etc.

process areas selected include software project planning, software project monitoring and control, software acquisition management, software requirements management, software quality assurance, software configuration management, software measurement and analysis, and software risk management.

About the Capability Maturity Model Integration

The Capability Maturity Model Integration (CMMI®) is an evaluation and appraisal model, developed by the Software Engineering Institute (SEI) at Carnegie Mellon University, which is used to evaluate the "maturity" of an organization's processes [3], [17]. The CMMI® models build on, extend, and integrate the best practices of the Capability Maturity Model for Software (SW-CMM®), the Systems Engineering Capability Maturity Model (SE-CMM®), and the Integrated Product Development Capability Maturity Model (IPD-CMM®) [18]. The model is defined in terms of Process Areas and Maturity Levels and has two representations:

- Staged representation that organizes the process areas by maturity level
- Continuous representation that organizes the process areas by process categories and measures capability level.

Although CMMI® was developed to reduce the risk of software procurements for the DOD, this model has become a popular framework for process improvement in both government and industry. Benefits of implementing recommended practices include significant improvements in software defects and consequent rework, cost and schedule predictability, and productivity of the development team. A list of the CMMI® Process Areas (PAs) is shown by maturity level in Table 3, and brief descriptions are given in Table 4.

JPL is currently implementing the CMMI® in four pathfinder software-intensive sections. The process areas they selected constitute those that are part of CMMI® Maturity Level 2 plus one from CMMI® Maturity Level 3. The near-term goal is to demonstrate CMMI® Maturity Level 2 compliance for selected projects in these sections by the end of fiscal year 2005. JPL's CMMI® target profile showing expected capability level for each process area by fiscal year (FY) is shown in Figure 1. Typical experience with the CMMI® model indicates that it takes, on average, about two years per maturity level, depending on the size of the organization involved. Progress will be assessed against the continuous representation of the CMMI® model. Formal CMMI® appraisals use the Standard CMMI® Appraisal Method for Process Improvement called SCAMPI. Informal, Class B appraisals, which use slightly more relaxed criteria for evidence, have been conducted at JPL for the past three years. Complete information on the CMMI® and SCAMPI can be found on the SEI's CMMI® website at <http://www.sei.cmu.edu/cmmi/cmmi.html>.

Available Products

In the past two years, the SQI Project developed a broad range of products, services, and training to support managers and practitioners throughout the software development life-cycle. Available products fall into the following categories:

1. Institutional requirements (includes policies, processes, procedures, and standards)
2. Compliance matrices
3. Handbooks and guides
4. Checklists
5. Templates
6. Sample documents
7. Studies and Reports (including engineering models)

Actual products available within each category are shown in Table 5. Each of these products was designed to assist managers and practitioners in generating the typical deliverable products that are part of the software life-cycle, and to ensure that those products comply with the JPL standard software process. This includes cost estimates, plans, reviews, documentation, test procedures, etc.

Some of the more popular products tend to be the document templates because they not only provide a document outline, but also include actual document format and content suggestions. The most helpful template has proven to be the Software Management Plan (SMP) template since it assists projects in planning their development activities. Training is discussed in Section 7, and consulting services are discussed in Section 8. Of course, as products, services and training were developed, each one needed to be systematically deployed.

Table 3 CMMI® Process Areas By Maturity Level (Staged Representation)

Maturity Level	Focus	CMMI® Process Areas	Category
5 Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment (OID) Causal Analysis and Resolution (CAR)	Adv. Process Mgmt. Adv. Support
4 Quantitatively Managed	Quantitative Management	Organizational Process Performance (OPP) Quantitative Project Management (QPM)	Adv. Process Mgmt. Adv. Project Mgmt.
3 Defined	Process Standardization	Requirements Development (RD) Technical Solution (TS) Product Integration (PI) Verification (VER) Validation (VAL) Organizational Process Focus (OPF) Organizational Process Definition (OPD) Organizational Training (OT) Integrated Project Management for IPPD (IPM) Risk Management (RSKM) Integrated Teaming (IT) Integrated Supplier Management (ISM) Decision Analysis and Resolution (DAR) Organizational Environment for Integration (OEI)	Engineering Engineering Engineering Engineering Engineering Basic Process Mgmt. Basic Process Mgmt. Basic Process Mgmt. Adv. Project Mgmt. Adv. Project Mgmt. Adv. Project Mgmt. Adv. Project Mgmt. Adv. Support Adv. Support
2 Managed	Basic Project Management	Requirements Management (REQM) Project Planning (PP) Project Monitoring and Control (PMC) Supplier Agreement Management (SAM) Measurement and Analysis (MA) Process and Product Quality Assurance (PPQA) Configuration Management (CM)	Engineering Basic Project Mgmt. Basic Project Mgmt. Basic Project Mgmt. Basic Support Basic Support Basic Support
1 Initial			

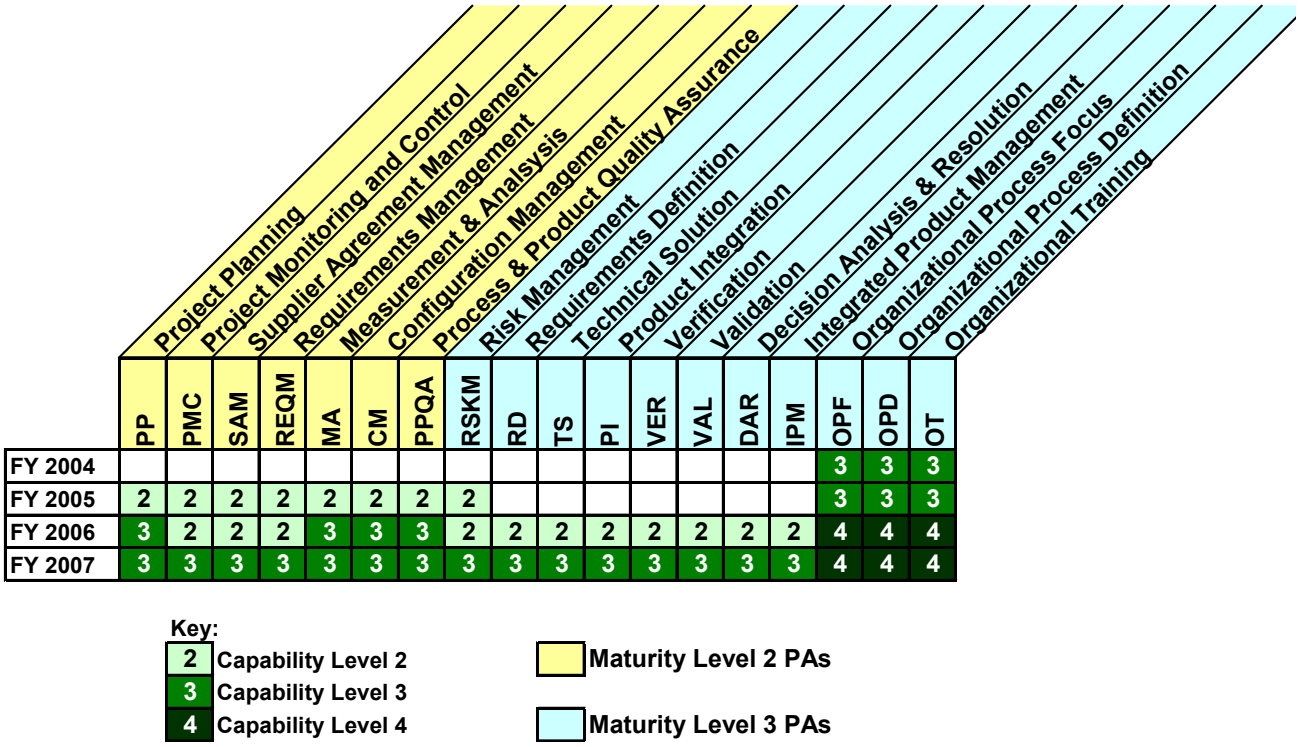

Figure 1 JPL CMMI® Target Profile (Capability Level by Fiscal Year)

Table 4 Brief Descriptions of CMMI® Process Areas (alphabetically)

CMMI Process Area	Acronym	Process Purpose
Causal Analysis & Resolution	CAR	Identify causes of defects and other problems and take action to prevent them from occurring in the future
Configuration Management	CM	Establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.
Decision Analysis and Resolution	DAR	Analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.
Integrated Project Management	IPM	Establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes. (For IPPD, also establish a shared vision for the project, and a team structure for integrated teams that will carry out the objectives of the project.)
Integrated Supplier Management	ISM	Proactively identify sources of products that may be used to satisfy the project's requirements and to manage selected suppliers while maintaining a cooperative project-supplier relationship.
Integrated Teaming	IT	Form and sustain an integrated team for the development of work products.
Measurement & Analysis	MA	Develop and sustain a measurement capability that is used to support management information needs.
Organizational Environment for Integration	OEI	Provide an integrated product and process development (IPPD) infrastructure, and manage people for integration.
Organizational Innovation & Deployment	OID	Select and deploy incremental and innovative improvements that measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives derived from the organization's business objectives.
Organizational Process Definition	OPD	Establish and maintain a usable set of organizational process assets.
Organizational Process Focus	OPF	Plan and implement organizational process improvement based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.
Organizational Process Performance	OPP	Establish and maintain a quantitative understanding of the performance of the organization's set of standard processes in support of quality and process-performance objectives, and provide process performance data, baselines, and models to quantitatively manage the organization's projects.
Organizational Training	OT	Develop the skills and knowledge of people so they can perform their roles effectively and efficiently.
Process and Product Quality Assurance	PPQA	Provide staff and management with objective insight into processes and associated work products.
Product Integration	PI	Assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.
Project Monitoring & Control	PMC	Provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan
Project Planning	PP	Establish and maintain plans that define project activities.
Quantitative Project Management	QPM	Quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives.
Requirements Development	RD	Produce and analyze customer, product and product-component requirements.
Requirements Management	REQM	Manage the requirements of the project's products and product components and identify inconsistencies between those requirements and the project's plans and work products.
Risk Management	RSKM	Identify potential problems before they occur, so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.
Supplier Agreement Management	SAM	Manage the acquisition of products from suppliers for which there exists a formal agreement.
Technical Solution	TS	Design, develop and implement solutions to requirements. Solutions, designs and implementations encompass products, product components, and product-related life-cycle processes either singly or in combinations as appropriate.
Validation	VAL	Demonstrate that a product or product component fulfills its intended use when placed in its intended environment.
Verification	VER	Ensure that selected work products meet their specified requirements

Use of Organizational Change Management (OCM)

Aware of the danger that these products could become "shelfware," JPL's senior management was eager to commit the resources to ensure that these changes were deployed into the software community, and that they impacted the way that software is managed, developed and acquired. As a result, the SQI Project took a very proactive approach to both organizational change management and customer relationship management. This approach is the antithesis of the typical one affectionately known as "If we build it, they will come." Instead, it involves proactively reaching out to customers, and doing whatever it takes to facilitate their understanding and usage of processes, products and services.

Organizational change management (OCM) is "the methodology that integrates change and the ability to adapt into the organization." [18] OCM involves working with a target community to systematically introduce them to desired changes in such a way that those changes are eventually adopted and become commonplace [11], [19]. It is based largely on Rogers' seminal work on diffusion of innovation [15] which is summarized in Table 6. The SQI Project has employed several of Rogers' strategies for diffusing innovations including relevance, customer focus, user friendliness, education, likelihood, measurement and testimony [1].

One of the premises of OCM is that people tend to fall into one of five change adoption categories and respond accordingly, as shown in Table 7. Another premise is that individuals tend to commit to change in predictable, sequential stages, beginning with contact and moving through awareness, understanding, installation or "trial use", adoption, institutionalization, and finally, internalization [6]. These stages are shown along the OCM curve in Figure 2. The SQI Project developed specific descriptions for what each of the OCM stages would mean at JPL and then utilized them in their OCM approach. See Table 8 for their OCM stage definitions and activities. They chose four pathfinder software-intensive sections to be "early adopters" and worked closely with them to implement various aspects of the CMMI® and the JPL software processes. In addition, they proactively reached out to projects in the early stages of development, e.g., Phase A – Mission and System Definition and Phase B – Preliminary Design. Also, they developed several venues for reaching out to the software community as a whole. These are described in more detail in Section 6.

4. DEPLOYMENT PROCESS

The SQI Project developed and is following a rigorous process for creating and deploying an asset that includes:

1. Collect user requirements and/or CMMI® needs.
2. Generate the process, product or artifact.

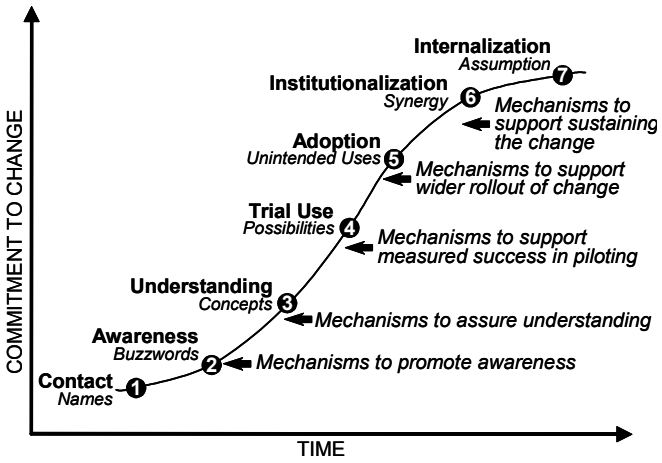

Figure 2 OCM Curve and Stages [6]

3. Develop Infrastructure and Operations approach and tools to support it.
4. Develop Communications and Outreach materials to support it.
5. Develop Education and Training materials to support it.
6. Perform Project Support to promulgate it.
7. Collect process and customer metrics to track it.
8. Capture and document Lessons Learned.
9. Update the process, product or artifact based on feedback.

Hence, the deployment process includes the following four parts which occur sequentially whenever a product is deployed, as shown in Figure 3:

1. Infrastructure and Operations -- develop the necessary infrastructure and operations approach for each area
2. Communication and Outreach – communicate with, and systematically reach out to, the user community so that they know what is available and understand where to obtain it.
3. Education and Training – provide classroom and computer-based training in the desired processes, products and tools
4. Project Support – provide consulting support to projects across a broad range of relevant topics.

These four parts of deployment are described in more detail in the next four sections of this paper.

5. INFRASTRUCTURE AND OPERATIONS

Infrastructure and Operations involves developing the necessary infrastructure and operations approach for the SQI Project as a whole, and also for each item to be deployed. It includes the contact management system, problem management system (action item tracking, problem/failure reporting), configuration management

system, electronic

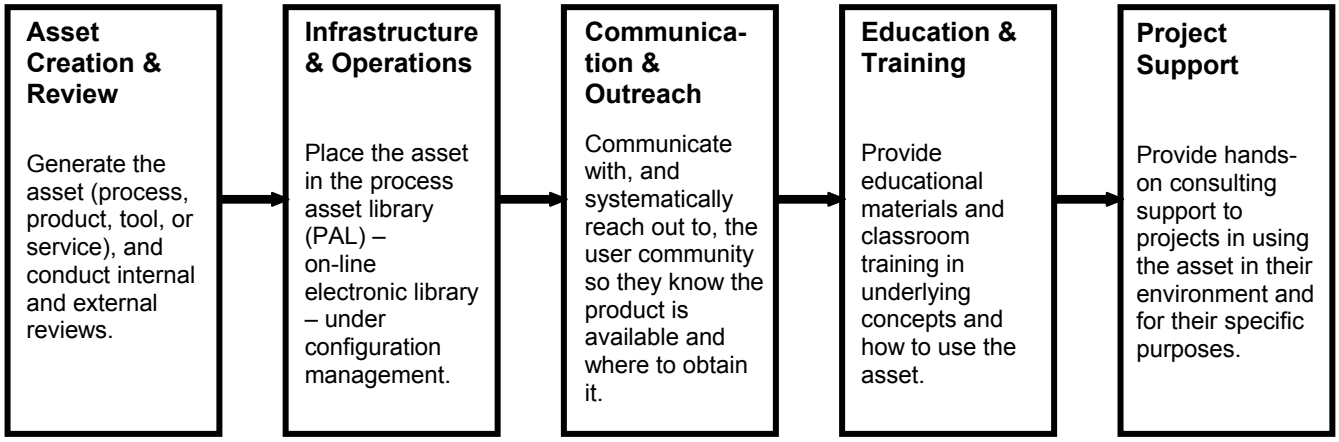

Figure 3 SQI Deployment Process

library, metrics collection, customer e-mail lists, customer tracking database for CRM, project calendar, target audience definition and strategy, intellectual property approach, and Operations Plan. This infrastructure is the foundation upon which all other aspects of deployment are built. Of course, the products and services themselves must first be generated, and this infrastructure greatly assists that process.

6. COMMUNICATION AND OUTREACH

Communication and Outreach involves communicating with, and systematically reaching out to, the user community so that they know and understand what is available. It includes a website, presentations, seminars, brochure, OCM and CRM approaches, surveys, forums, interest groups, etc.

The SQI Project generated an SQI OCM approach that defined the following:

- Organizational change management strategy
- Infusion goals and change acceptance time lines
- Themes and thrusts
- Key stakeholders and segments
- Communications vehicles
- SQI logo, brochure, fliers, and tag lines
- Roles and responsibilities, and interactions amongst the elements to achieve the changes.

This OCM approach informs all aspects of their communication and outreach activities and provides a well-integrated approach to their customers.

Similarly, the SQI Project generated an SQI CRM approach that defined the primary, secondary and tertiary customer target segments and corresponding customer profiles. Then they tailored specific communication messages and approaches to the needs of each customer segment.

The SQI Project developed an extensive website to support their user community, and gave the site a URL that is very easy to remember. The website is structured along the lines of the product categories in Table 5. It also includes information about training, seminars, CMMI, frequently asked questions, contacts, etc. They also generated a tailorable presentation describing the products, services and training available, and are in the process of giving the presentation to all software-intensive organizations at the lab. In addition, they developed a 3-fold brochure, bookmark and cubicle clip to help promote their website and services. They use multiple communication channels to communicate their message, ranging from community e-mail lists, to websites, posters, fliers, cafeteria monitors, newsletters, etc.

Also, the SQI Project sponsors several seminars in an attempt to reach out to the software community as a whole, and to a particular set within that, namely Software Test Engineers. These seminars are described below.

1. SQI Software Seminar Series – Shares information about the practices and methodologies for improving software quality. The series consists of one-hour noon-time presentations by internal JPL speakers on various software topics concerned with software processes, practices, methodologies, and project experiences.

2. SQI Software Tool Service (STS) Seminar Series – Highlights software engineering tools available from industry and academia. The series consists of short seminars and tutorials offered by various vendors on their commercial-off-the-shelf (COTS) tools for use in the software development process, including CASE tools, operating systems, languages, debugging tools, and test tools, etc.

3. JPL Software Test Guild – Provides a forum for JPL Test Engineers to network, learn and share knowledge. The guild consists of one-hour special interest group (SIG) meetings of software test engineers covering topics ranging from test tools to test methodologies to lessons learned about testing in various application domains and test environments.

Table 6 Rogers' Diffusion of Innovation Model

Attributes of Innovation	Strategies for Optimizing Attributes
Relative Advantage – degree to which the innovation is perceived to improve upon existing solutions	**Technology Improvement** – Introduce a new technology that is more powerful than the existing technology.
Compatibility – the difficulty associated with mastering the new innovation	**Relevance** – Make sure the problem solved by the innovation is important to adopters. **Realism** – Do not try to change too much at once or to please too many different types of users. **Customer Focus** – Seek input from current and future adopters and design a solution that they want.
Complexity – the difficulty associated with mastering the new innovation	**Developer Friendliness** – Reduce the learning curve for developers of the innovation. **User Friendliness** – Reduce the learning curve for adopters by making the innovation easy to learn and use. **Reuse** – Reuse as much of the old process and technology as possible. **Education** – Provide tutorials and demonstrations to potential users and managers. Publish useful information on Web pages and offer pointers to Early Adopters.
Trialability – the ability to experiment with the innovation before adopting it in normal operations	**Cost** – Reduce the cost of trial use. **Likelihood** – Increase the likelihood that trial use will succeed.
Observability – the ease with which improvement is noticed after adoption of the innovation	**Measurement** – Collect data about the old and new technologies for comparison. **Testimony** – Provide forums for adopters to describe their experiences. **Shadowing** – Provide a side by side comparison by running two projects with the same goals, but with one using the old technology and the other using the new.

Table 7 Rogers' Categories of People and Their Responses to Innovation

Categories	Characteristics	Responses to Innovation
Innovators or "Techies" – those who create new technologies	Gatekeepers for any new technology; appreciate technology for its own sake; appreciate architecture of technology; will spend hours trying to get technology to work; very forgiving of poor documentation, slow performance, incomplete functionality, etc.; helpful critics	Will settle for buggy or difficult-to-use solution components; are accustomed to finding their way around the glitches.
Early Adopters or "Visionaries" – those who are the first to try innovations	Dominated by a dream or vision; focus on business goals; usually have close ties with "techie" innovators; match emerging technologies to strategic opportunities; look for breakthrough; thrive on high visibility, high risk projects; have charisma to generate buy-in for projects; do not have credibility with early majority	Can see the strategic advantage of the improvement or change and are willing to help the organization get there.
	"The Chasm" in the Adopter Continuum	
Early Majority or "Pragmatists" – those who establish an innovation's success by adopting it for regular use	Do not want to be pioneers (prudent souls); control majority of budget; want percentage improvement (incremental, measurable, predictable progress); not risk averse, but want to manage it carefully; hard to win over, but are loyal once won.	Can see the advantage of the improvement or change and are willing to carefully adopt it.
Late Majority or "Conservatives" – those who adopt an innovation after its success has been demonstrated	Avoid discontinuous improvement (revolution); adopt only to stay on par with the rest of the world; somewhat fearful of new technologies; like pre-assembled packages with everything bundled.	Need a lot of support to adopt the solution component.
Laggards or "Skeptics" – those who never adopt or who do so reluctantly after it becomes necessary	"Nay sayers"; adopt only after technology is not recognizable as separate entity; constantly point at discrepancies between what was promised and what is	Are very resistant to changing the status quo, despite the effectiveness of the solution component.

Types of Products	Products Available in Each Category
1. Institutional Requirements	Flight Project Practices (FPPs) Design, Verification/Validation and Operations Principles Software Development Requirements (SDR) Policy Flight Project Gate Documents

Table 8 SQI OCM Stages and Activities

OCM Stages	OCM Stage Name	OCM Definition at JPL	Associated OCM Activities
0	**None**	Never heard of SQI Project	None
1	**Contact**	Have heard of SQI Project	SQI publicity and outreach activities -- SQI brochure, bookmark, cubicle clip, announcements, e-mail, fliers, posters
2	**Awareness**	Aware of SDRs, Software website, SQI Road Show, and existence of SQI Software Seminar Series	SDRs and FPPs in JPL Rules!, Software website, SQI "Road Show", SQI Software Seminar Series, user forums, surveys
3	**Understanding**	Understand SDRs, basic SQI products (templates, handbooks, guides, etc.) and CMMI® Maturity Level 2 (ML 2) Process Areas (PAs)	SDR Awareness Briefing, SDR Overview course, SQI training courses – Software Mgmt./Engr. & Process Improvement; SQI Software Seminar Series, Software Test Guild
4	**Installation (Trial Use)**	Utilize SDRs and some SQI products and services; implement specific practices of some CMMI® PAs (CL 1)	SQI consulting –planning, SDRs, cost estimates metrics, tools, etc.; benefits & rationale, case studies, SQI impact metrics
5	**Adoption**	Some orgs/projects comply with institutional policies and practices (SDRs and FPPs); implement some CMMI® PAs at Capability Level 2 (CL 2) -- specific and generic practices	Target sections and Process Engineers., SQI Rep./Shepherd, CMMI® Class B appraisals & SCAMPI, CMMI® Implementation Plans, more training/coaching, lessons learned; address barriers to change.
6	**Institutionalization**	All mission-critical software orgs and projects comply with institutional policies and practices (SDRs and FPPs); achieve CMMI® Maturity Level 2/3	SQI Element activities, CMMI® Profile, CMMI® Class B appraisals and formal SCAMPI, SQI OCM activities and metrics
7	**Internalization**	All software orgs and projects comply with institutional policies and practices (SDRs and FPPs); achieve CMMI® Maturity Level 3	CMMI® Profile, CMMI® Class B appraisals and formal SCAMPI, continuous process improvement, comprehensive training program, reward and recognition system

7. EDUCATION AND TRAINING

Education and Training involves providing educational materials and classroom and computer-based training in the desired processes, products and tools. It includes a JPL Software Training Plan that defines the target customers, required skills sets, and training goals, and describes the training process to be utilized [23]. Courses are offered on such topics as software management, software engineering, and process improvement [10]. A biannual training survey is conducted in order to gauge the impact of, and satisfaction with, the software training program. In addition, the four-part Kirkpatrick Model is used to evaluate training effectiveness [13]. (See Table 9 for a description of JPL's use of this model.) The course titles and target audience in each category are discussed below.

Software Management Courses

Currently four software management courses are offered to Project Managers (PMs) and Project Element Managers (PEMs) to give them a general overview of software project planning, and then more details on software project monitor and control. All courses are offered quarterly and include Software Management and Planning (SMP), Quantitative Software Management (QSM), Software Risk Management (SRM), and Software Acquisition Management (SAM). A course under development provides an overview of the Develop Software Process (DSP) and of the software development requirements (SDR), i.e., the policy and guidelines.

Software Engineering Courses

Currently six software engineering courses are offered to Cognizant Engineers (Cog Es) and Software Engineers, including Software Product Engineering (SPE), Software Peer Reviews (SPR), Software Architecture, Software Testing, System Software Reliability, and System Requirements and Management. The last course covers systems requirements as well as software requirements.

Software Process Improvement Courses

The software process improvement training is focused on the Capability Maturity Model Integration (CMMI®). Four courses are offered including Overview of CMMI®, Introduction to CMMI®, Intermediate CMMI®, and Mastering Process Improvement. While these courses are

Table 9 SQI Use of the Kirkpatrick Model for Evaluating Training Effectiveness

Kirk-patrick Model Level	Kirkpatrick Model Level Name	Kirkpatrick Model Definition at JPL	Associated Kirkpatrick Model Activities
1	**Reactions**	Collect reactions of class attendees, and perceptions of instructors and SQI Training Coordinator	Completed module and course evaluation forms from course participants, evaluation compilation and analysis, verbal feedback from instructors, Training Coordinator perception of reception
2	**Learning**	Determine what principles, facts and techniques were understood and absorbed by the class attendees, i.e., what they now know and are able to do as a result of the training	Results of follow-up interviews conducted <=10 weeks after class, annual training survey results; demonstrated comprehension of course concepts and techniques, pre- and post- test vehicles and results
		OCM and CMMI® Activities Below	
3	**Behavior**	Determine changes in "on-the-job" behavior in the workplace itself related to the desired and taught behavior vs. original behavior.	Course objectives, specific desired behaviors for each module, specific practices of CMMI® PAs; detailed behavior surveys, "Quick Look" evaluations, SQA assessments, CMMI® Class B Appraisals
4	**Results**	Determine specific results across the workplace, i.e., progress towards primary SQI goals: cost and schedule predictability, quality of mission-critical software, project start-up time, productivity, defect rates, and reuse of software products.	SQI measurement program, especially institutional trends; foundation models, measures, and databases.

primarily meant for SQI Project personnel, process engineers, system engineers, and any others involved in process improvement, it is recommended that senior managers and other managers also take the overview course.

8. PROJECT SUPPORT

Project Support involves providing consulting support to projects across a broad range of relevant topics so that they can use the products in their own environment and for their specific purposes. It includes consulting in the areas of cost estimation, software project planning, software project tracking, earned value management (EVM), metrics definition and implementation, defects and reliability, software acquisition, software tools, use of templates, software testing, software quality assurance, CMMI®, etc.

Extensive consulting support is provided to the four target sections and also to projects in the early phases of the system life-cycle. Additional ad hoc consulting is provided as requested. Examples of consulting support provided to projects include:

1. Support for generating software cost estimates (effort, schedule, budget) for proposals, Phase A studies or detailed Phase B cost estimates, based on the Software Cost Estimation Handbook, the Cost Database, Software Cost Analysis Tool (SCAT) [2], and "rules of thumb" provided in various institutional models (e.g., productivity, development effort by phase, schedule time by effort, etc.)
2. Support for doing software project planning, especially generating a Software Management Plan (SMP), based on the SMP Template, the Software Development Requirements (SDRs) policy, the Software Process Tailoring Guide, the Software Risk Management Handbook, sample documents, etc.
3. Support for evaluating, selecting, and procuring various software engineering tools via the Software Tool Service (STS).
4. Support for generating various types of documentation based on the various document templates, applicable handbooks, sample documents, etc. (Figure 4 shows how a user can start with basic project information and the appropriate document template, then apply information from a handbook or follow the example shown in a sample document, and utilize consulting to generate a completed document more quickly.)

All customer contacts made by SQI are tracked via the SQI contact management system and categorized as follows: Information, Outreach, Training, and Consulting. This consulting support is provided free of charge up to a certain point, usually around 40 hours.

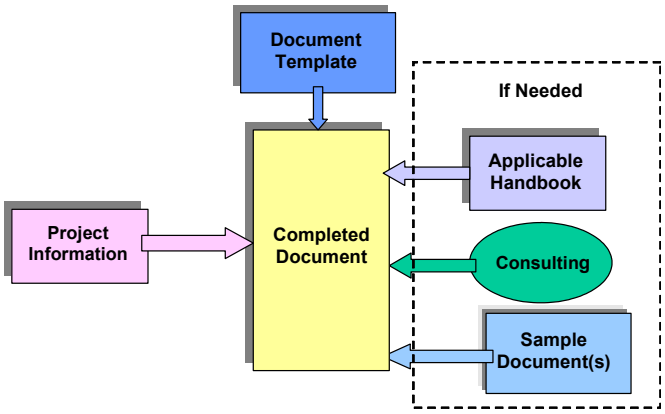

Figure 4 Software Documentation Using SQI Templates, Handbooks and Sample Documents

9. LESSONS LEARNED

The JPL SQI Project has collected a number of observations or "lessons learned" from its efforts to improve the management of software in an engineering and scientific environment. These observations or lessons fall into three basic categories: process improvement, creating assets, and deploying assets.

Process Improvement

1. Start with a proven framework. – The CMMI® framework offered a proven process improvement approach and appraisal benchmark. It facilitated the measurement of progress against that benchmark.
2. Get outside help if necessary. – Consulting support on CMMI® provided by appraisers from the Center for Systems Management (CSM) proved to be very helpful in understanding the model and in generating evidence for appraisals. The advice of the Lead Appraiser to "start small, start slow, and start simple" [12] helped in setting realistic goals and objectives, and establishing an appropriate horizon for change acceptance.
3. Build on previous efforts. – The SQI Project was able to build on previous reengineering efforts, especially the major reengineering activities associated with Process-Based Management (PBM) and ISO 9000 in the late 1990's.
4. Reach the "front line" managers too. – The SQI Project is very fortunate to have the "must haves" necessary for success in creating major change (as described in Section 1). Senior management support is important, however, so is the support of "front line" managers (supervisors). That support needs to be painstakingly earned, one meeting or presentation at a time.
5. Other concurrent major changes can be a mixed blessing, i.e., sources of distraction or opportunity. [9] – Recently the entire Engineering and Science Directorate (ESD) at JPL underwent a major reorganization. The effects on the software improvement effort still need to be fully understood. It means many new players, but also provides many additional opportunities.

Creating Assets

6. Start by documenting the current processes. – Documenting the current processes provides a necessary baseline and a basis for future improvement.
7. Provide tools to support requirements and facilitate process compliance. – When you begin with specific process requirements and then develop tools to actually implement them, e.g., handbooks, templates, and models, then users are much more likely to comply and to perceive the process as "user friendly."
8. Utilize many reviewers to promote ownership. – When many reviewers who are representative of different domains and perspectives provide comments on new or revised products, it promotes ownership or "buy-in" of the final result.
9. Strive for consistency amongst products – When many products and guidelines are generated over time by several different individuals, there can be issues with consistency amongst those products. At some point, it is necessary to step back and review the entire product suite to ensure that they are consistent with the standards and frameworks being utilized, and that they are consistent with each other. That is, standards, handbooks, templates and training about a particular topic all need to convey the same message.
10. Allocate sufficient time for curriculum development. – Curriculum development and defining course content is very time consuming. On average, it takes approximately four months to develop a new course.
11. Update courses regularly. – When processes and products are changing and the environment is dynamic, courses need to be updated regularly to reflect the latest information.

Deploying Assets

12. Use OCM and CRM to facilitate change. – There are several advantages to proactively using OCM and CRM. It helps to maintain a customer focus and to create motivation for reaching out to customers. Also, when setting priorities, it helps to know who your primary target audience is vs. your secondary or tertiary audience. For example, some "eager beavers" or early adopters may not be part of your primary customer group.
13. Pair Process Engineers with "Shepherds". – Process Engineers from each of the target sections were paired with a representative or "shepherd" from the SQI Project who worked with them to understand CMMI® and its implications for their organization.
14. Address "culture issues" head on. – The SQI Project needed to address some culture issues it encountered when deploying assets.
    a. The major difference between how software is

developed for Flight Software applications and how it is developed for Ground Software applications, especially software that is developed for multi-mission purposes.

b. Factors that engendered resistance to change, such as the perception of insufficient time and resources to try something new, program and project constraints, and the difficulty of change itself.

c. "Baggage" from previous process improvement efforts and the false perception of "just another unfunded mandate."

15. Conduct internal appraisals periodically. – When periodic internal appraisals are conducted, it reveals progress and helps to show what additional effort is required to meet the stated goals.
16. Collect metrics and measure regularly. – Collect metrics, and conduct regular surveys and user forums to determine the level of infusion into the organization and to uncover any barriers to acceptance. Measure infusion, effectiveness, customer satisfaction, progress, etc. Remember that "without measurement, you're just guessing!" [14]
17. Track customer contacts. – It is helpful to track customer contacts so that there are no duplications, and customers don't feel pestered by many representatives covering the same territory. It is also useful for reporting outreach efforts to management.
18. Use an electronic library. – Use of an electronic library promotes information sharing and collaboration among various team members and projects.
19. Communicate via multiple avenues and promote shamelessly. It never helps a change effort if it remains a "well kept secret." No matter how many times a message is communicated, there still may be some who haven't gotten the message. Some people prefer e-mail, while others prefer fliers, posters, presentations or seminars. It helps to think of novel ways to attract attention such as bookmarks, cubicle pins, brochures, websites, etc.
20. Address frequently asked questions about products and about who to contact. Create a set of frequently asked questions (FAQs) to reflect recent enquiries and to anticipate the types of concerns customers may have. Make available a list of Points of Contact (POCs) for various types of issues and subject matter experts for each area.

10. CONCLUSIONS

Changes in how software is managed, developed and acquired do not come quickly or easily. The improvement process needs to be approached with many of the same deliberate methods and practices that are used in actual system development [3], [8]. It helps to proactively reach out to customers instead of merely waiting for them to come to you. Lastly, it is important to maintain the proper balance between defining processes or generating assets and actually deploying them and supporting customers. If this balance is not achieved, all the products generated just become "shelfware."

11. ACKNOWLEDGEMENTS

Many people have contributed to the success of JPL's Software Quality Improvement Project and deserve recognition.

- Thomas R. Gavin – Associate Director for Flight Projects and Mission Success
- Chris Jones, William Weber, John Beckman – Directors for program and engineering Directorates
- David Nichols – Process Owner for the Develop Software Products (DSP) process
- David Swenson – Manager of the Project Support Office and the Technical Infrastructure (TI) Council
- Frank Kuykendall, Chi Lin, and Steve Flanagan – SQI Project Management, past and present.
- Milton Lavin – SQI Project Engineer
- Jairus Hihn – SQI Measurement, Estimation and Analysis Element
- John Hackney and Pam Francel – SQI Process and Product Definition Element
- Kathleen Crean, George Rinker, Jody Steinbacher, Bill Taber, and Blake Whittington – Software Process Engineers
- Alan Ferdman – CMMI Internal Appraisal Team Lead
- Brian Vickers – SQI Software Training and Seminar Coordinator
- John Greensage – SQI Deployment Systems Engineer
- Michelle Medina – SQI Software Training Logistics Support
- Ray Kile and Carlo Rodriquez – CMMI® Lead Appraisers from CSM, evaluation of JPL's CMMI® Maturity Level 2/3 Process Areas (PAs).

In addition several instructors spent time developing course content and instructing classes: Robert Barry, Erich Corduan, Dan Crichton, William Decker, David Eisenman, Dan Erickson, Alan Ferdman, John Hackney, Dave Hermsen, Jairus Hihn, Terry Himes, Suzanne Klein, Frank McGarry, Ron Morillo, George Rinker, Nicolas Rouquette, Kimberly Simpson, Tuyet-Lan Tran, and former instructors Milton Lavin, Burt Sigal and Jody Steinbacher. Nor should we forget ~1000 students who have attended our courses and provided valuable feedback.

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration (NASA). Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government, NASA or the Jet Propulsion Laboratory, California Institute of Technology.

12. ACRONYMS AND ABBREVIATIONS

CL	Capability Level
CMMI	Capability Maturity Model Integration
COTS	Commercial Off-The-Shelf
CRM	Customer Relationship Management
DSMS	Deep Space Mission System
DSP	Develop Software Products
FPP	Flight Project Practices
FY	Fiscal Year
IS&CS	Information Systems & Computer Science
JPL	Jet Propulsion Laboratory
ML	Maturity Level
NASA	National Aeronautics & Space Administration
OCM	Organizational Change Management
PA	Process Area
SCAMPI	Standard CMMI® Appraisal Method for Process Improvement
SDR	Software Development Requirements
SEI	Software Engineering Institute
SMP	Software Management Plan
SQI	Software Quality Improvement
TQM	Total Quality Management

REFERENCES

[1] Mark A. Ardis and Janel A. Green, *Successful Introduction of Domain Engineering into Software Development,* Bell Labs Technical Journal, July-Sept. 1998

[2] Barry W. Boehm, et al, *Software Cost Estimation with COCOMO II,* Prentice-Hall PTR, Prentice-hall Inc., Upper Saddle River, New Jersey, 2000

[3] Mary Beth Chrissis, Mike Konrad, Sandy Shrum, *CMMI: Guidelines for Process Integration and Product Improvement,* Addison-Wesley, San Francisco, 2003

[4] William H. Davidow and Bro Uttal, *Total Customer Service: The Ultimate Weapon,* Harper Perennial, 1990

[5] Stewart Deck, *What Is CRM?*, CIO.com, May 2001

[6] SuZ Garcia and Chuck Myers, *Out from Dependency: Thriving as an Insurgent in a Sometimes Hostile Environment*, SEPG Conference, 2001

[7] Randy Harris, *What Is a Customer Relationship Management (CRM) System?*, DarwinMag.com, December 2003

[8] Watts S. Humphrey, *A Discipline for Software Engineering,* Addison-Wesley, New York, 1995

[9] David H. Hutton, *The Change Agents' Handbook: A Survival Guide for Quality Improvement Champions*, ASQ Quality Press, Milwaukee, Wisconsin, 1994

[10] P. A. "Trisha" Jansma, *Got Software? What Managers and Engineers Need to Know,* IEEE Aerospace Conference, March 2004

[11] Jerald M. Jellison, *Overcoming Resistance: A Practical Guide to Producing Change in the Workplace*, Simon and Schuster, New York, 1993

[12] Ray Kile, *Starting a Process Improvement Program Effectively*, Center for Systems Management, Inc., 2003

[13] Donald L. Kirkpatrick, *Evaluating Training Programs: The Four Levels*, Berrett-Koehler Publishers, Inc., 2nd edition, July 1998

[14] David F. Rico, *ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers*, J. Ross Publishing, Boca Raton, Florida, 2004

[15] Everett M. Rogers, *Diffusion of Innovations, 5th Edition,* Free Press, 2003

[16] Mohanbir Sawhney, *Fundamentals of Value*, CIO.com, July 2003

[17] *Capability Maturity Model Integration, Version 1.1,* Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 2002

[18] Dallas Independent School District, *Project OASIS: Managing Organizational Change*, Oracle Applications User Group: Software Change Management SIG, Spring Conference 2002

[19] *Journal of Organizational Change Management* ISSN: 0953-4814

[20] *NASA Program and Project Management Processes and Requirements,* NASA Procedures and Guidelines, NPG 7120.5B, 2002

[21] *NASA Software Management, Engineering, and Assurance,* NASA Procedures and Guidelines, NPG 2820

[22] *People Capability Maturity Model, Version 2.0,* Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 2001

[23] U.S. Office of Personnel Management, Office of Workforce Relations, *A Guide to Strategically Planning Training and Measuring Results,* OWR-35, July 2000

BIOGRAPHY

P. A. "Trisha" Jansma is the Project Element Manager (PEM) for the Deployment Element of the Software Quality Improvement (SQI) Project at the Jet Propulsion Laboratory (JPL) in Pasadena, California. With over 30 years at JPL in both line and task management positions, she has a broad background in systems and software engineering and information technology, in engineering and scientific environments. Jansma has extensive experience in the management, design, development and delivery of cost-effective, software-intensive systems. She has experience in all facets of project life-cycle development, from initial feasibility analysis, proposal development and conceptual design through documentation, implementation, user training, enhancement and operations. Jansma has a B.A. in Mathematics from Point Loma Nazarene University, an M.S. in Computer Science from the University of Southern California, and an Executive M.B.A. from the Peter F. Drucker Graduate School of Management at Claremont Graduate University. She also holds a California Community College Teaching Credential and a California Secondary Teaching Credential, and has taught Systems and Software Engineering courses at the graduate level.