How To Measurably Improve Your Requirements

NDIA Systems Engineering Conference
October 2007

Timothy G. Olson, President
Lean Solutions Institute, Inc.
(760) 804-1405
Tim.Olson@lsi-inc.com
www.lsi-inc.com

Objectives

Describe some requirements problems from industry.

Present a useful classification of requirements problems.

Describe some practical strategies and best practices that organizations have used to successfully develop, manage, and improve their requirements in a measurable way.

Provide real examples that address requirements problems.

Answer any of your questions.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI® Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Why Focus on Requirements?

The hardest single part of building a system is deciding what to build...
No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

Adapted from Fredrick Brooks, Jar. [Brooks 87]

Why Focus on Requirements?

A research report from the Standish Group highlighted the continuing quality and delivery problems in our industry and identified three leading causes:

- Lack of user input
- Incomplete requirements and specifications
- Changing requirement specifications

- Reference: "Chaos", Compass, The Standish Group, 1997, used with permission.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Problems with Requirements

According to the SEI [Christel 92], problems of requirements elicitation can be grouped into 3 categories:

1. Problems of Scope: the requirements may address too little or too much information.

2. Problems of Understanding: problems within groups as well as between groups such as users and developers.

3. Problems of Volatility: the changing nature of requirements.

Scope and Volatility

The list of 10 requirements elicitation problems given in [McDermid 89] can be classified according to the 3 categories in [Christel 92]:

Problems of Scope

- The boundary of the system is ill-defined
- Unnecessary design information may be given

Problems of Volatility

- Requirements evolve over time

Problems of Understanding

- Users have incomplete understanding of their needs
- Users have poor understanding of computer capabilities and limitations
- Analysts have poor knowledge of problem domain
- User and analyst speak different languages
- Ease of omitting “obvious” information
- Conflicting views of different users
- Requirements are often vague and untestable, e.g., “user friendly” and “robust”

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Requirements Management (REQM)

SG 1: Manage Requirements:

SP 1.1-1: Obtain an Understanding of the Requirements

SP 1.2-2: Obtain Commitment to Requirements

SP 1.3-1: Manage Requirements Changes

SP 1.4-2: Maintain Bidirectional Traceability of Requirements

SP 1.5-1: Identify Inconsistencies between Project Work and Requirements

- Reference: "Capability Maturity Model® Integration (CMMI), Version 1.1", CMU/SEI-2002-TR-011, March 2002

Requirements Development (RD)

SG 1: Develop Customer Requirements:

SP 1.1-1: Collect Stakeholder Needs

SP 1.1-2: Elicit Needs

SP 1.2-1: Develop the Customer Requirements

SG 2: Develop Product Requirements:

SP 2.1-1: Establish Product and Product-Component Requirements

SP 2.2-1: Allocate Product-Component Requirements

SP 2.3-1: Identify Interface Requirements

SG 3: Analyze and Verify Requirements:

SP 3.1-1: Establish Operational Concepts and Scenarios

SP 3.2-1: Establish a Definition of Required Functionality

SP 3.3-1: Analyze Requirements

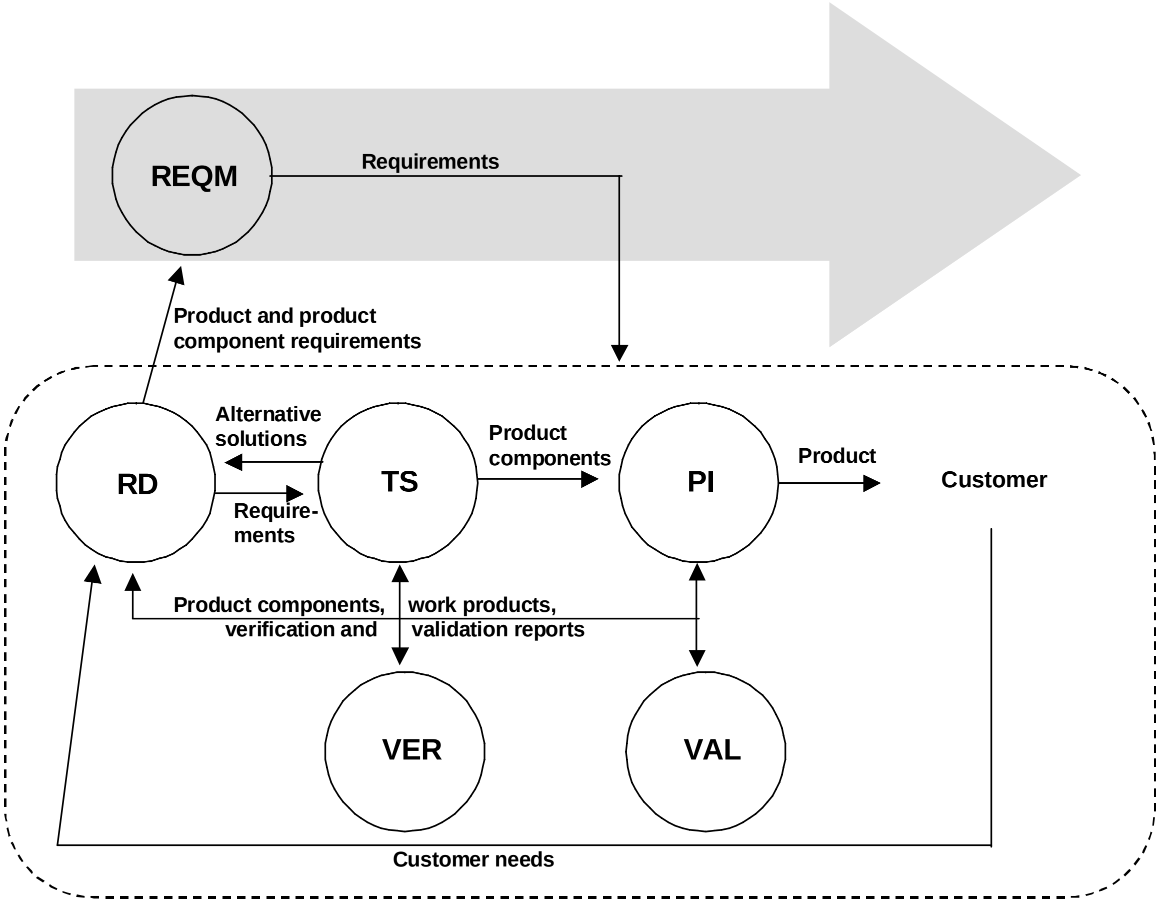
SP 3.4-3: Analyze Requirements to Achieve Balance

SP 3.5-1: Validate Requirements

SP 3.5-2: Validate Requirements with Comprehensive Methods

- Reference: "Capability Maturity Model® Integration (CMMI), Version 1.1", CMU/SEI-2002-TR-011, March 2002

Engineering Process Areas



- Reference: "Capability Maturity Model® Integration (CMMI), Version 1.1", CMU/SEI-2002-TR-011, March 2002

CMMI and Requirements

Requirement processes need to be defined, trained, and improved (e.g., OPF, OPD, OT, OID).

Support processes are critical for measuring and managing requirements (e.g., CM, MA, PPQA).

Defects need to be removed and prevented in requirements (e.g., PI, VER, VAL, CAR).

IPPD (i.e., integrated product teams) also contains allocating requirements to teams (e.g., IPM).

Supplier Sourcing requires managing supplier requirements (e.g., SAM).

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Practical Strategies

1. Define a lean Requirements Management (REQM) Process.

2. Use lean Configuration Management (CM) and CM Metrics.

3. Use Requirements Metrics (e.g., priority, stability, risk, number of requirements, defect density, etc).

4. Define the requirements process (RD), and use lessons learned from quality (e.g., QFD, Juran, etc).

5. Tailor a requirements standard (e.g., IEEE).

6. Use early defect detection and defect prevention.

7. Use operational definitions to define requirements.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

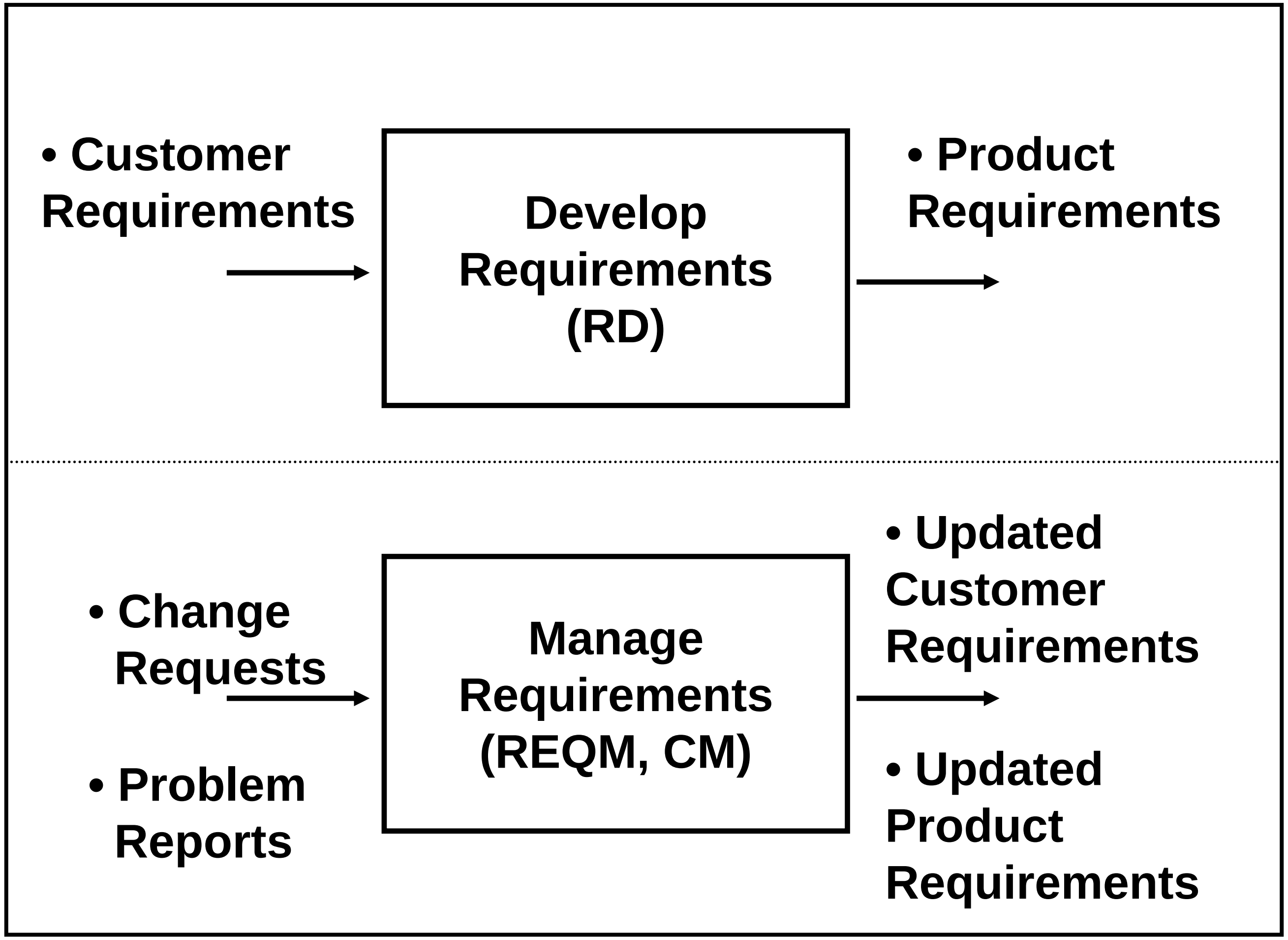
Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

1. Define Lean Requirements Processes (REQM, RD)



- Customer Requirements

Develop Requirements (RD)

- Product Requirements

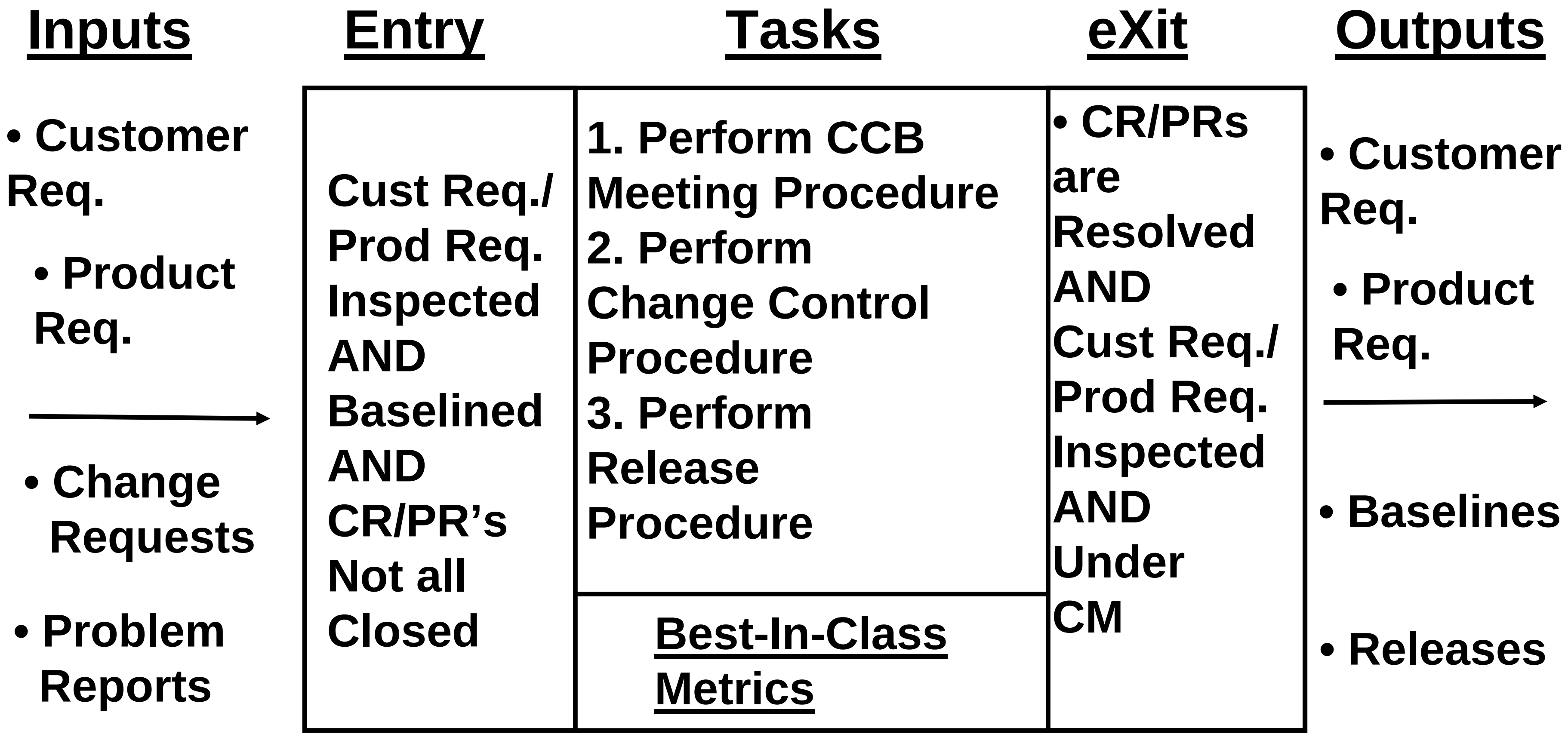
- Change Requests
- Problem Reports

Manage Requirements (REQM, CM)

- Updated Customer Requirements
- Updated Product Requirements

1. Manage Requirements (REQM)

Purpose: Effectively Manage Requirements Changes



Inputs

- Customer Req.
- Product Req.
- Change Requests
- Problem Reports

Entry

Cust Req./ Prod Req. Inspected AND Baselined AND CR/PR's Not all Closed

Tasks

1. Perform CCB Meeting Procedure
2. Perform Change Control Procedure
3. Perform Release Procedure

Best-In-Class Metrics

eXit

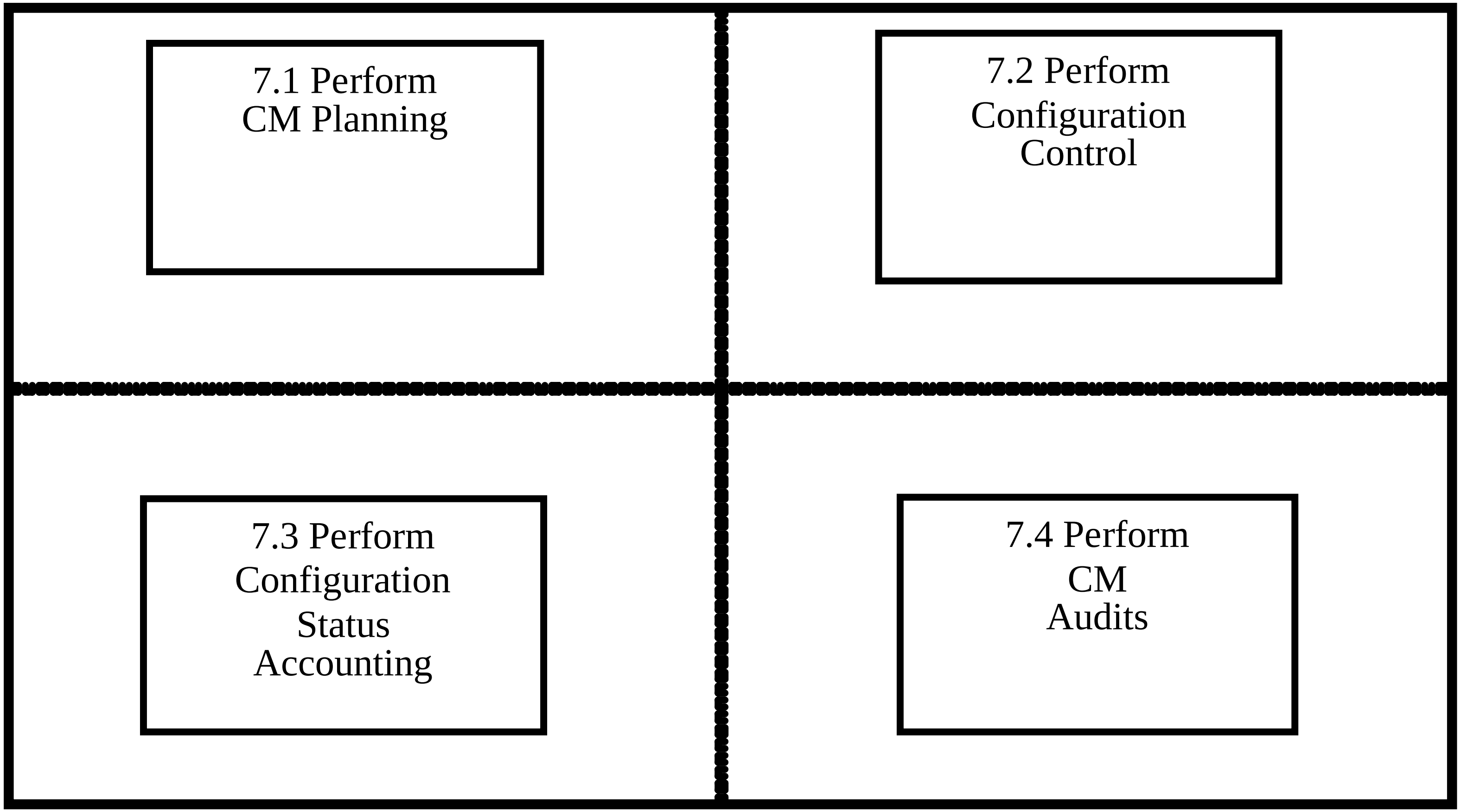
- CR/PRs are Resolved AND Cust Req./ Prod Req. Inspected AND Under CM

Outputs

- Customer Req.
- Product Req.
- Baselines
- Releases

Roles: Project Manager (PM), CCB

1. Example Lean NASA JPL MGSS CM Process



7.1 Perform CM Planning

7.2 Perform Configuration Control

7.3 Perform Configuration Status Accounting

7.4 Perform CM Audits

[Olson 2006a] Olson, Timothy G., “Defining a Lean CM Process at NASA JPL”, Presentation, NDIA CMMI Conference, November 2006.

1. Example Lean CM Process

5 W's on
1 Page
in a
Process
Model

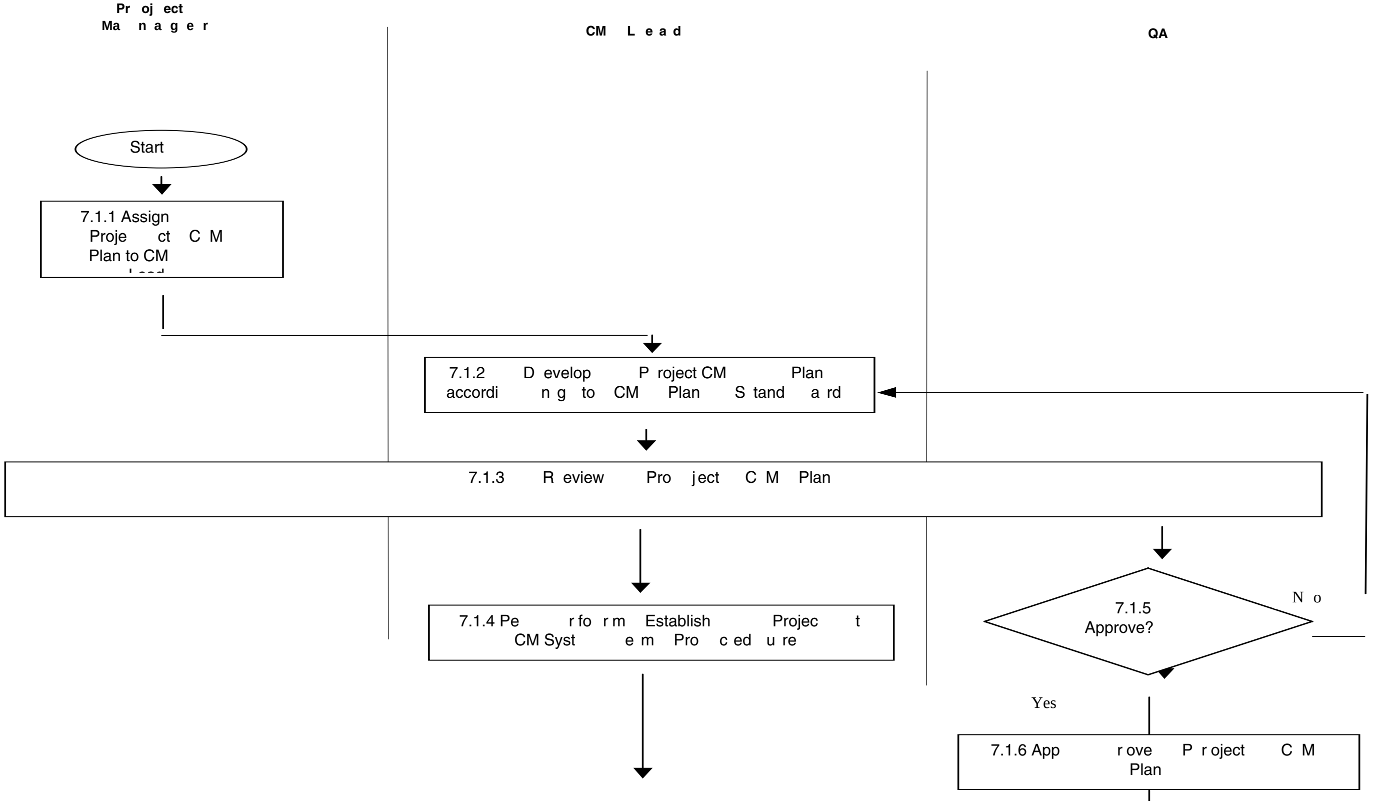
7.1 Perform CM Planning

PURPOSE: Develop CM Plans that meet project needs.

PROCESS

INPUTS/ENTRY CRITERIA:
CONTEXT
- Project initiated AND Draft Project Plan
- Organizational CM Plan is approved and meets CM Standard.

ROLES & ACTIVITIES:



Project Manager

CM Lead

QA

Start

7.1.1 Assign Project CM Plan to CM Lead

7.1.2 Develop Project CM Plan according to CM Plan Standard

7.1.3 Review Project CM Plan

7.1.4 Perform Establish Project CM System Procedure

7.1.5 Approve?

No

Yes

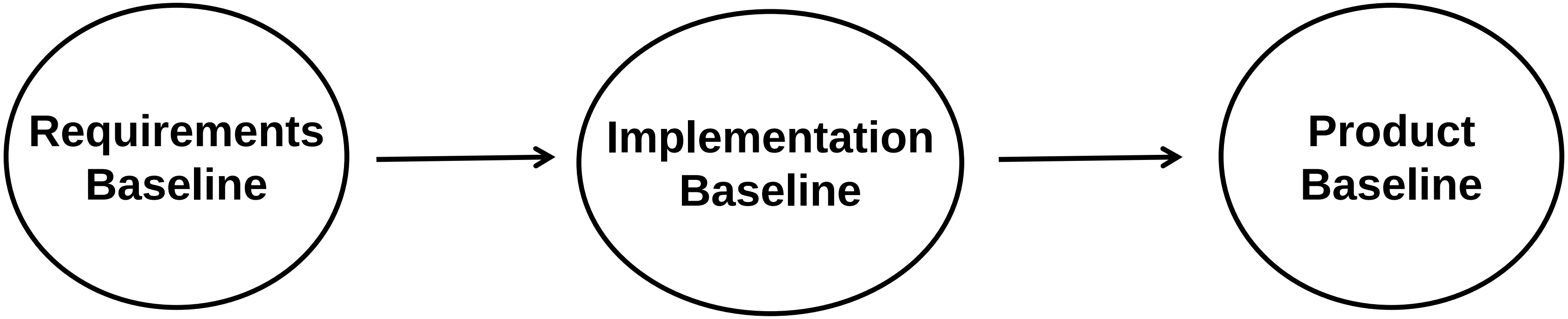
7.1.6 Approve Project CM Plan

Patent
Pending
Approach

[Olson 2006a] Olson, Timothy G., "Defining a Lean CM Process at NASA JPL", Presentation, NDIA CMMI Conference, November 2006.

2. Use CM and CM Metrics

Fundamental Baselines



Place the requirements under formal CM and use CCB's to control changes.

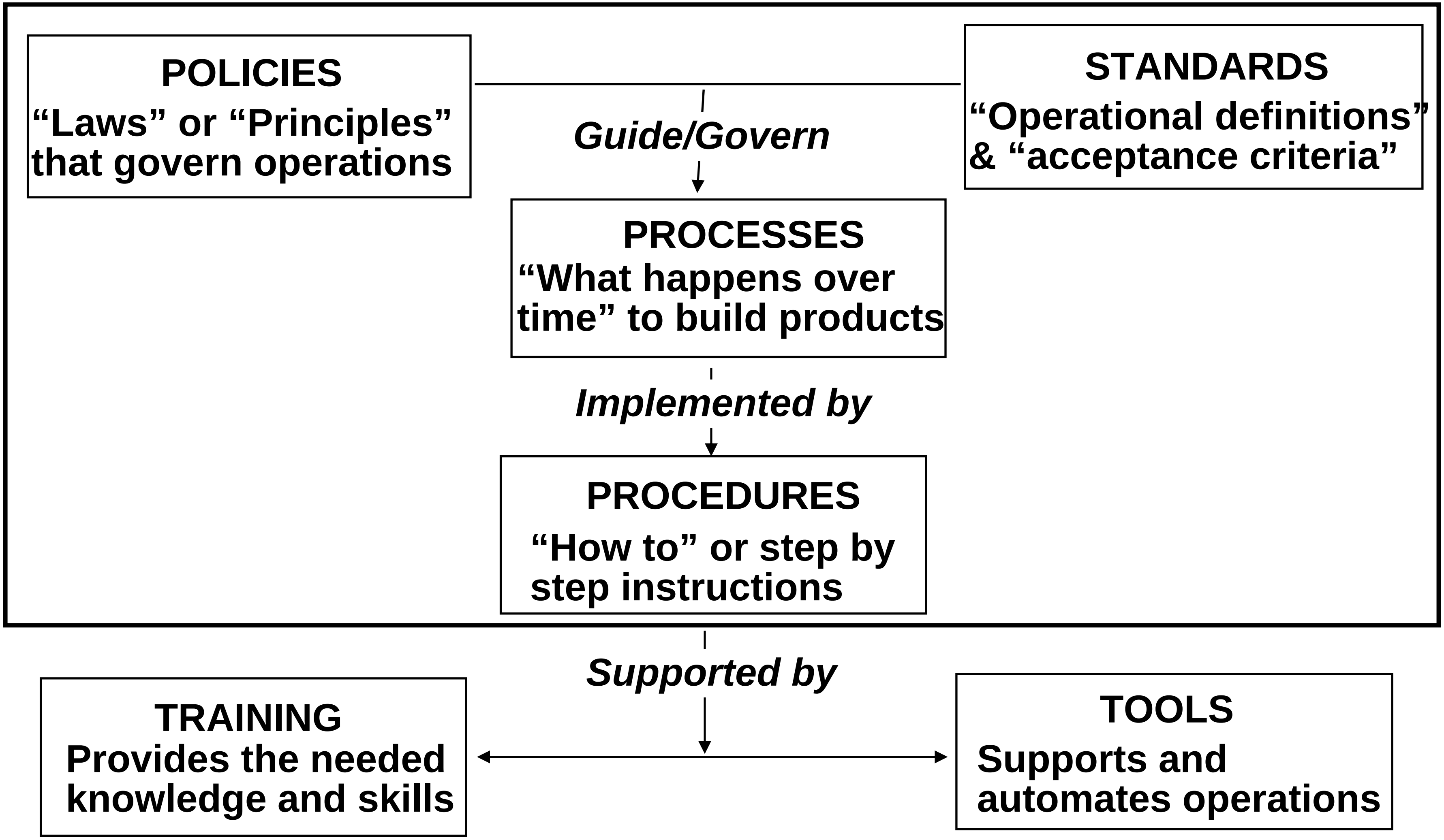
Example CM Metrics:

- Number of CRs/PRs (e.g., open vs. closed over time)
- Requirements Volatility (e.g., number of CRs per requirement)

3. Example Requirement Metrics

#	Requirement	Reference (e.g., customer)	Allocation	Stability (H/M/L)	Risk (H/M/L)	Priority (H/M/L)
1	System shall send an RTF FAX	SOW # 10-20.3	Software	H	L	M
2	Aircraft position shall be updated by the Inertial Navigation System (INS) Solution	ORD #2-30-20.3.4.4	Software	M	M	H

4. Documentation Framework



POLICIES
“Laws” or “Principles” that govern operations

STANDARDS
“Operational definitions” & “acceptance criteria”

Guide/Govern

PROCESSES
“What happens over time” to build products

Implemented by

PROCEDURES
“How to” or step by step instructions

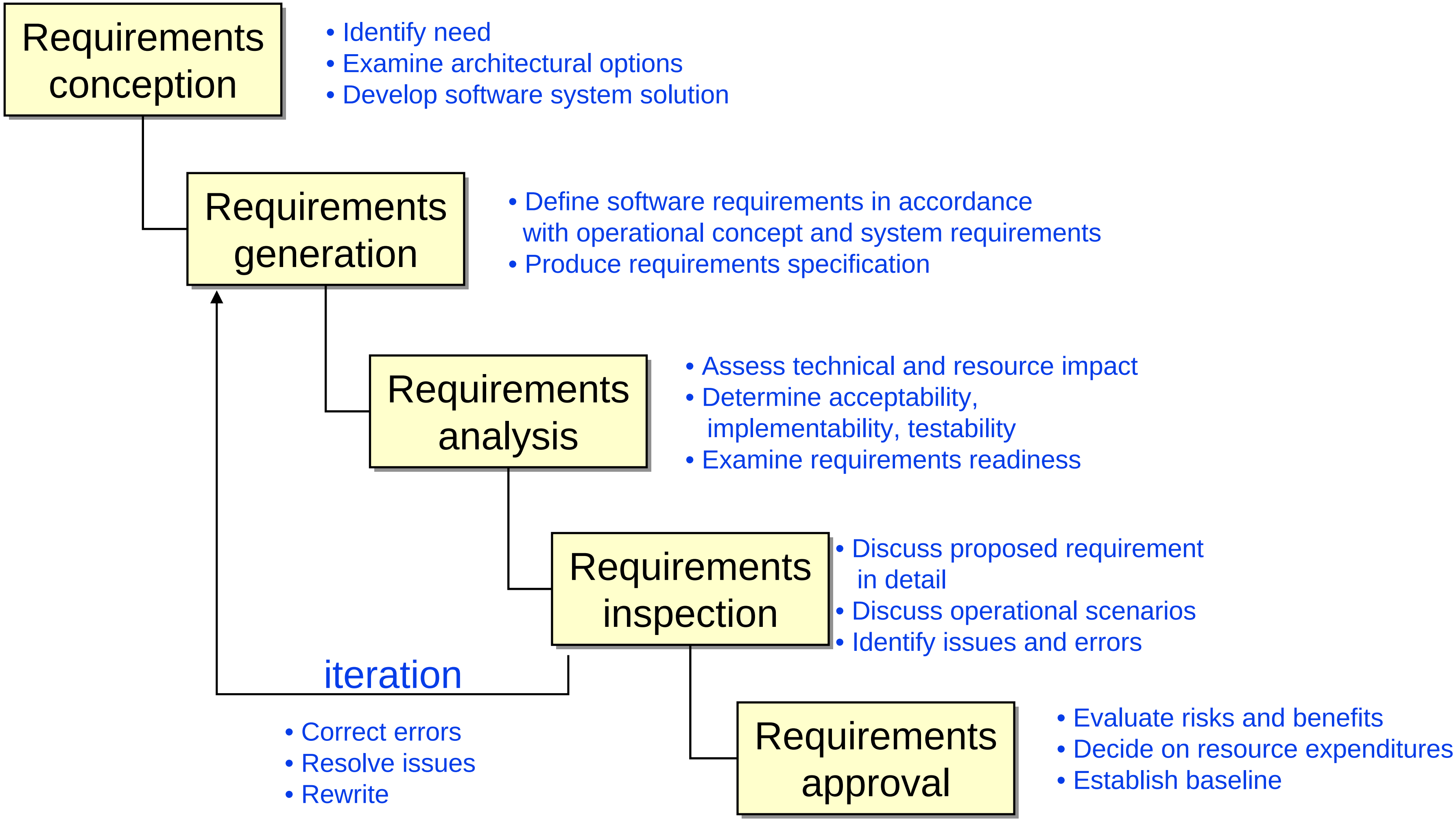
Supported by

TRAINING
Provides the needed knowledge and skills

TOOLS
Supports and automates operations

- Slide adapted from”A Software Process Framework for the SEI Capability Maturity Model”, CMU/SEI-94-HB-01

4. Requirements Process - NASA Onboard Shuttle Project



Requirements conception

- Identify need
- Examine architectural options
- Develop software system solution

Requirements generation

- Define software requirements in accordance with operational concept and system requirements
- Produce requirements specification

Requirements analysis

- Assess technical and resource impact
- Determine acceptability, implementability, testability
- Examine requirements readiness

Requirements inspection

- Discuss proposed requirement in detail
- Discuss operational scenarios
- Identify issues and errors

Requirements approval

- Evaluate risks and benefits
- Decide on resource expenditures
- Establish baseline

iteration

- Correct errors
- Resolve issues
- Rewrite

5. IEEE SyRS and SRS Standard Outlines

SyRS

1.0 Introduction
2.0 General System Description
3.0 System Capabilities, Conditions, and Constraints
- 3.1 Physical
- 3.2 System Performance Characteristics
- 3.3 System Security
- 3.4 Information Management
- 3.5 System Operations
- 3.6 Policy and Regulation
- 3.7 System Life Cycle

4.0 System Interfaces

- Adapted from: IEEE Std 1233-1998

SRS

1.0 Introduction
2.0 Overall Description
3.0 Specific Requirements
- 3.1 External Interface Requirements
- 3.2 Functional Requirements
- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.5 Software System Attributes
- 3.6 Other Requirements

Appendices
Index

- Adapted from: IEEE Std 830-1998

5. Organizing SRS Section 3

SRS Section 3 can be organized by:

- Mode
- User Class
- Object
- Feature
- Stimulus/Response
- Functional Hierarchy
- Multiple organizations

6. Example Requirements Checklist Categories

1. Clarity
2. Completeness
3. Complexity
4. Consistency
5. Constraints
6. Feasibility
7. Functionality/Logic
8. Interfaces
9. Standards
10. TBDs
11. Testability
12. Traceability

Etc.

7. Example Operational Definition

What is a good requirement? When is a requirement defined? Questions like these are difficult to answer without operational definitions.

An operational definition precisely and concisely defines a measurable requirement that states [NASA 96]:

- What it has to do
- How well it has to do it
- Under what conditions it has to do it

7. Example Operational Definition

#	Requirement (What)	Conditions	Upper Limit	Lower limit	Base Measure
1	Report total percentage of students that passed the first test and graduated	Students that pass first test by => 70% score	Calculate Percentage to 3 decimal places	Plus or minus .001	Percent
2	Report total percentage of students that failed the second test and did not graduate	Students that failed second test by < a 70% score	Calculate Percentage to 3 decimal places	Plus or minus .001	Percent

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Some Advanced Strategies

Juran Model: Customer requirements are written in the customer’s language, then translated into the product requirements written in producer’s language.

QFD/Juran’s Quality Planning Process: Measurable requirements that meet customer needs using a defined process (e.g., House of Quality).

Usage Scenarios/Use Cases/Operational Scenarios: A powerful way to identify requirements based on user needs.

Requirements written in formal languages.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Summary

The hardest single part of building a system is the requirements.

The top requirements problems are inadequate requirements specifications, changes to requirements, and lack of user input.

Requirements elicitation problems fall into problems of scope, understanding, and volatility.

There are practical strategies that you can use today that will help you address problems with requirements.

References

- [Billings 1994] Billings, C., et al. “Journey to a Mature Software Process”, IBM Systems Journal, vol. 33, no. 1, 1994.
- [Brooks 1987] Brooks, Fredrick P., Jar. “No Silver Bullet: Essence and Accidents of Software Engineering”. IEEE Computer, 10-19, April 1987.
- [Christel 1992] Christel, Michael G. and Kang, Kyo C. “Issues in Requirements Elicitation”, CMU/SEI-92-TR-12, 1992.
- [CMMI 02] “Capability Maturity Model® Integration (CMMI Version 1.1”, CMU/SEI-2002-TR-011, March 2002
- [Ebenau 1994] Ebenau, B. and Strauss, S., Software Inspection Process. McGraw-Hill, 1994.
- [Fagan 1986] Fagan, M. “Advances in Software Inspections”, M. IEEE Transactions on Software Engineering, July 1986
- [Gilb 1993] Gilb, T. and Graham, D. Software Inspection. Addison-Wesley, 1993.
- [Humphrey 1989] Humphrey, W. S. Managing the Software Process. Reading, MA: Addison-Wesley Publishing Company, 1989.
- [IEEE 1993] “IEEE Recommended Practice for Software Requirements Specifications”, IEEE Std 830-1993.
- [IEEE 1996] “IEEE Guide for Developing System Requirements Specifications”, IEEE Std 1233-1996.
- [Juran 1988] Juran, J. and Gryna, F. Juran’s Quality Control Handbook. McGraw-Hill, Fourth Edition, 1988.
- [Mays 1990] Mays, et al. “Experiences with Defect Prevention”. IBM Systems Journal, 1990.
- [McDermid 1989] McDermid, J.A. “Requirements Analysis: Problems and the STARTS Approach”, in IEE Colloquium on ‘Requirements Capture and Specification for Critical Systems’ (Digest no. 138), 4/1-4/4. Institution of Electrical Engineers, Nov. 1989.
- [McMenamin 1984] McMenamin, S. and Palmer, J. Essential Systems Analysis. Yourdon Press Computing Series, Prentice-Hall, 1984.
- [NASA 1996] Requirements Capture and Evaluation Process” Training Notebook, Lockheed Martin Space Information Systems (NASA Shuttle Onboard Software - SEI Level 5 Project), 1996.
- [Olson 2007a] Olson, Timothy G., “Measurably Improving Your Systems Engineering Requirements”, Presentation and Paper, INCOSE Systems Engineering Conference, June 2007.
- [Olson 2007b] Olson, Timothy G., “How to Define Good Service Requirements”, Tutorial, ASQ Service Quality Conference, October 2007.
- [Olson 2006a] Olson, Timothy G., “Defining a Lean CM Process at NASA JPL”, Presentation, NDIA CMMI Conference, November 2006.
- [Olson 2006a] Olson, Timothy G., “How To Measurably Improve Your Requirements”, Tutorial, NDIA Systems Engineering Conference, October 2006.
- [Olson 20031] Olson, Timothy G. “Successful Strategies for Improving Requirements”, ASQ 13th International Conference on Software Quality, Orlando, FL, 2003.
- [Olson 2003a] Olson, Timothy G. “Measurably Improving Your Requirements Based on the CMMI”, NDIA CMMI Conference, Denver, CO, 2003.
- [Olson 2002] Olson, Timothy G. “How to Practically Improve Your Requirements Process Using the CMMI Framework”, Presentation, NDIA CMMI Conference, Denver, CO, 2002.
- [SEI 1990] “Software Engineering Institute Affiliates Symposium 1990” proceedings. Pittsburgh, PA: Carnegie Mellon University, 1990.

How To Measurably Improve Your Requirements

NDIA Systems Engineering Conference
October 2007

Timothy G. Olson, President
Lean Solutions Institute, Inc.
(760) 804-1405
Tim.Olson@lsi-inc.com
www.lsi-inc.com