How to Practically Improve Your Requirements Process Using the CMMI^SM Framework

2002 CMMI^SM Technology Conference

Tim Olson, President
Quality Improvement Consultants, Inc. (QIC)
Juran Institute Associate
Authorized SEI Lead Assessor
(760) 804-1405
Tim.Olson@att.net

Dennis Beeson
QIC Associate
SEI Candidate Lead Assessor
(760) 375-3376
ddbeeson@gte.net

Objectives

Describe some requirements problems from industry.

Present a useful classification of requirements problems.

Describe some practical approaches that real organizations have used to successfully develop and manage their requirements.

Provide real examples that address requirements problems.

Answer any of your questions.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI^{SM} Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Why Focus on Requirements?

"The hardest single part of building a software 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."

Fredrick Brooks, Jr. [Brooks 87]

Why Focus on Requirements?

One study [Beichter 84] estimates that 60% of system errors are due to inadequate specification and design.

According to the SEI National Software Capacity Study [SEI 90], the top 2 out of 10 factors that contribute to the failure of system development contracts to meet schedule or costs are requirements problems (1 to 5 scale: 1=not serious; 3=serious; 5=very serious):

1. Inadequate requirements specification (4.5)
2. Changes in requirements (4.3)

Why Focus on Requirements?

A recent 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?

$\boxed{\text{A Practical Requirements Classification}}$

CMMI^{SM} 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. <u>Problems of Scope:</u> the requirements may address too little or too much information.

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

3. <u>Problems of Volatility:</u> 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^{SM} Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Requirements Management (RM)

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^{SM}), 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:

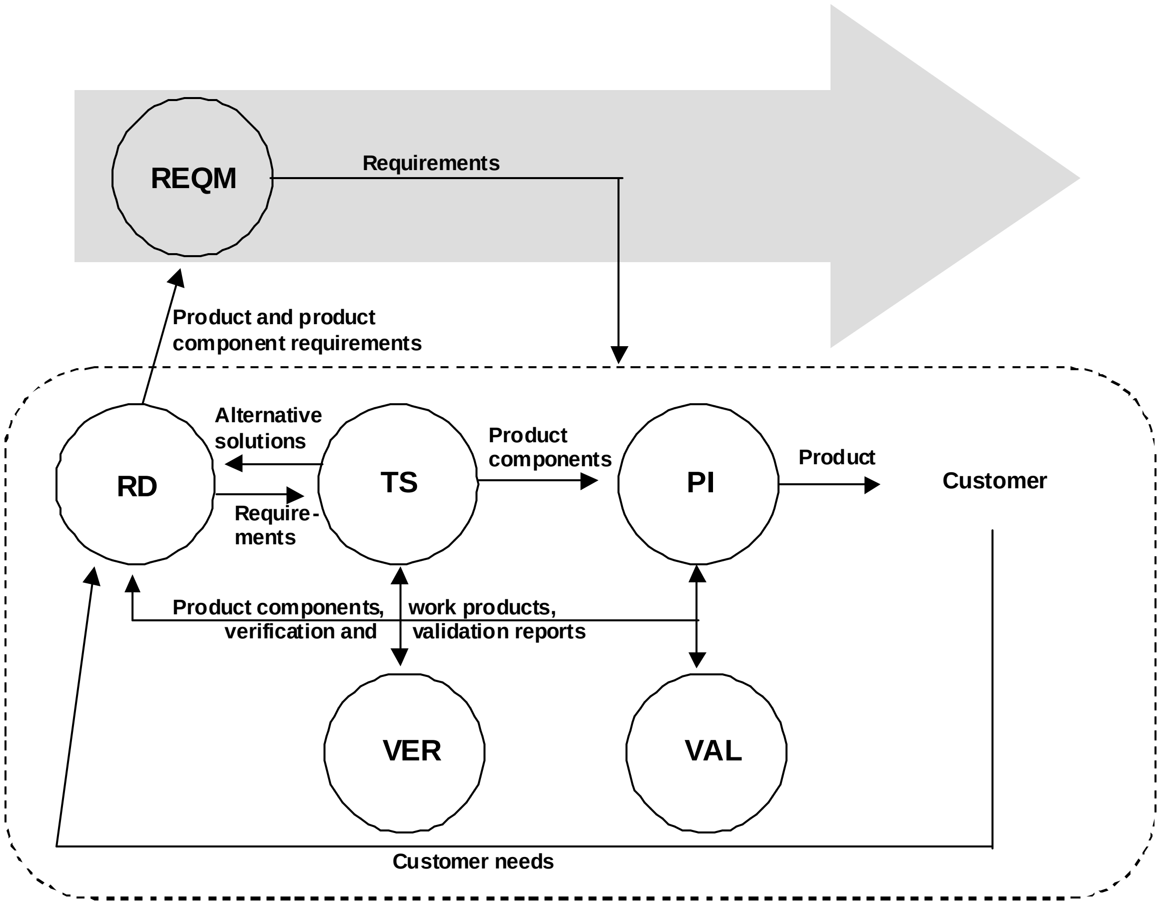
SP 2.1-1: Establish Product and Product-Component Requirements
SP 2.2-1: Allocate Product-Component Requirements
SP 2.3-1: Identify Interface Requirements

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


REQM
Requirements
Product and product component requirements
RD
Alternative solutions
Require-ments
TS
Product components
PI
Product
Customer
Product components, work products, verification and validation reports
VER
VAL
Customer needs

• Reference: “Capability Maturity Model® Integration (CMMI^{SM}), Version 1.1”, CMU/SEI-2002-TR-011, March 2002

CMMI^{SM} 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 also contains allocating requirements to teams (e.g., IPM for IPPD).

Supplier Sourcing requires managing supplier requirements.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI^{SM} Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Practical Approaches

1. Use CMMI^{SM} or SW-CMM® requirements management (REQM or RM)

2. Use configuration management (CM)

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

4. Define the requirements process using best practices.

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

6. Use inspections and defect prevention.

7. Use operational definitions to define requirements.

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI^{SM} Requirements Overview

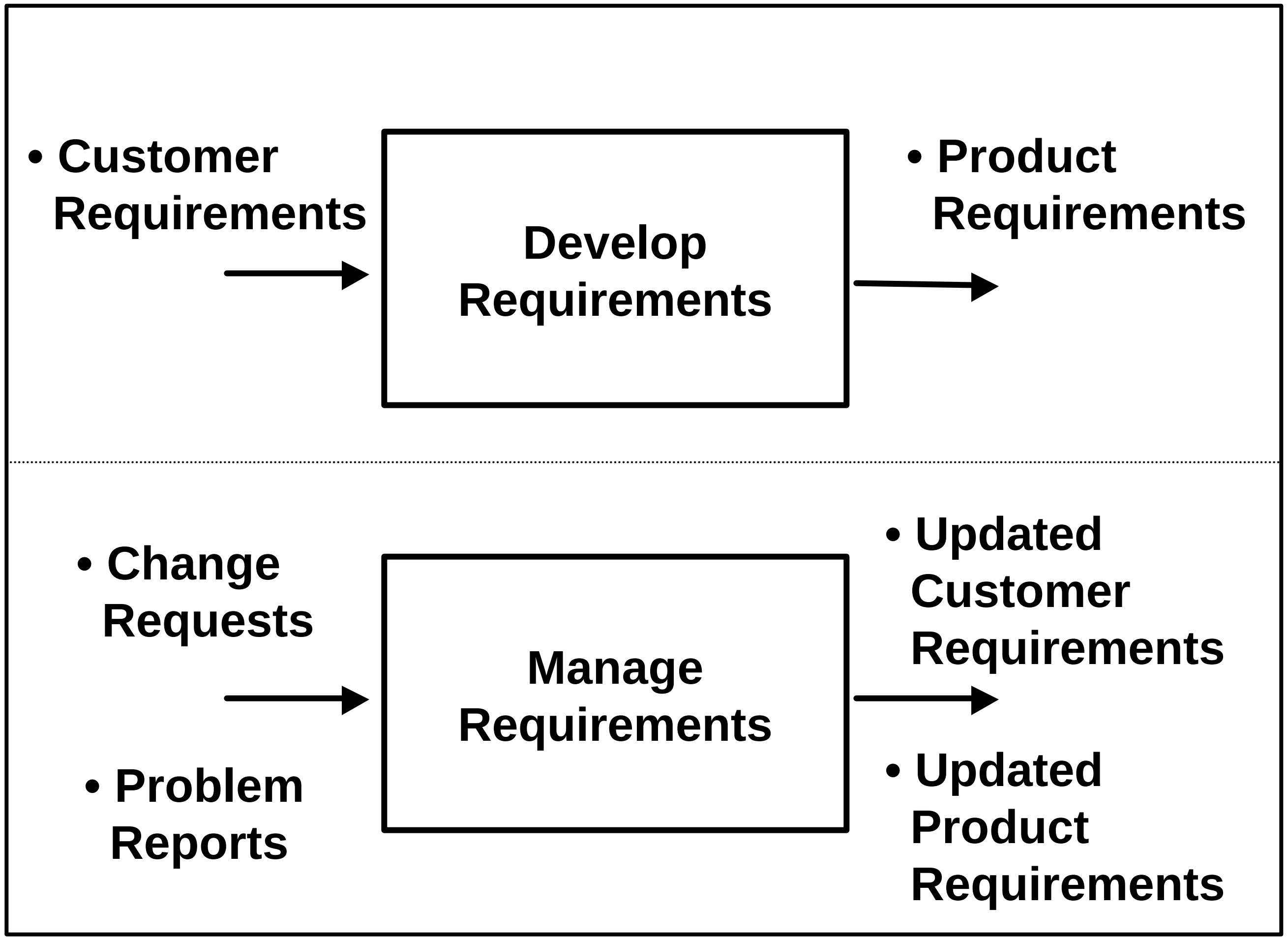
Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

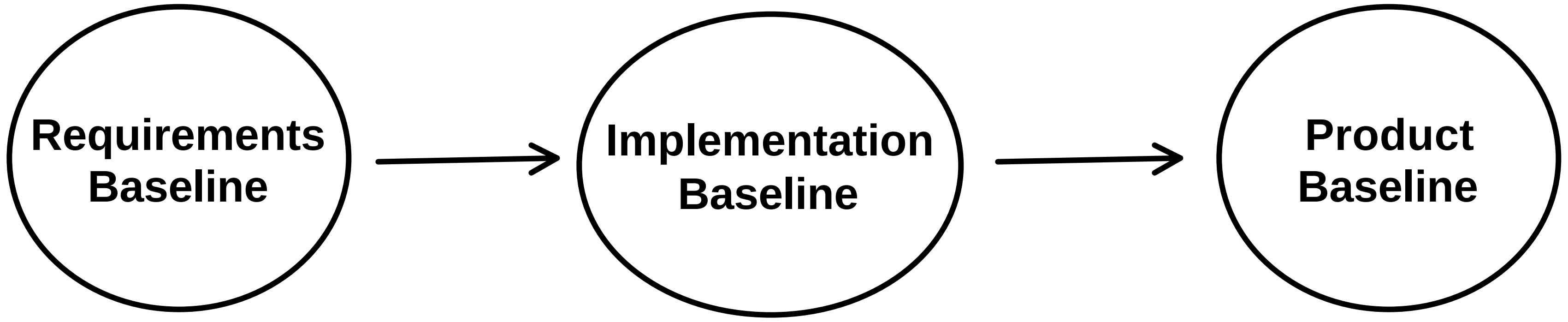
Summary

1. Requirements Management Process



2. Use CM

Fundamental Baselines

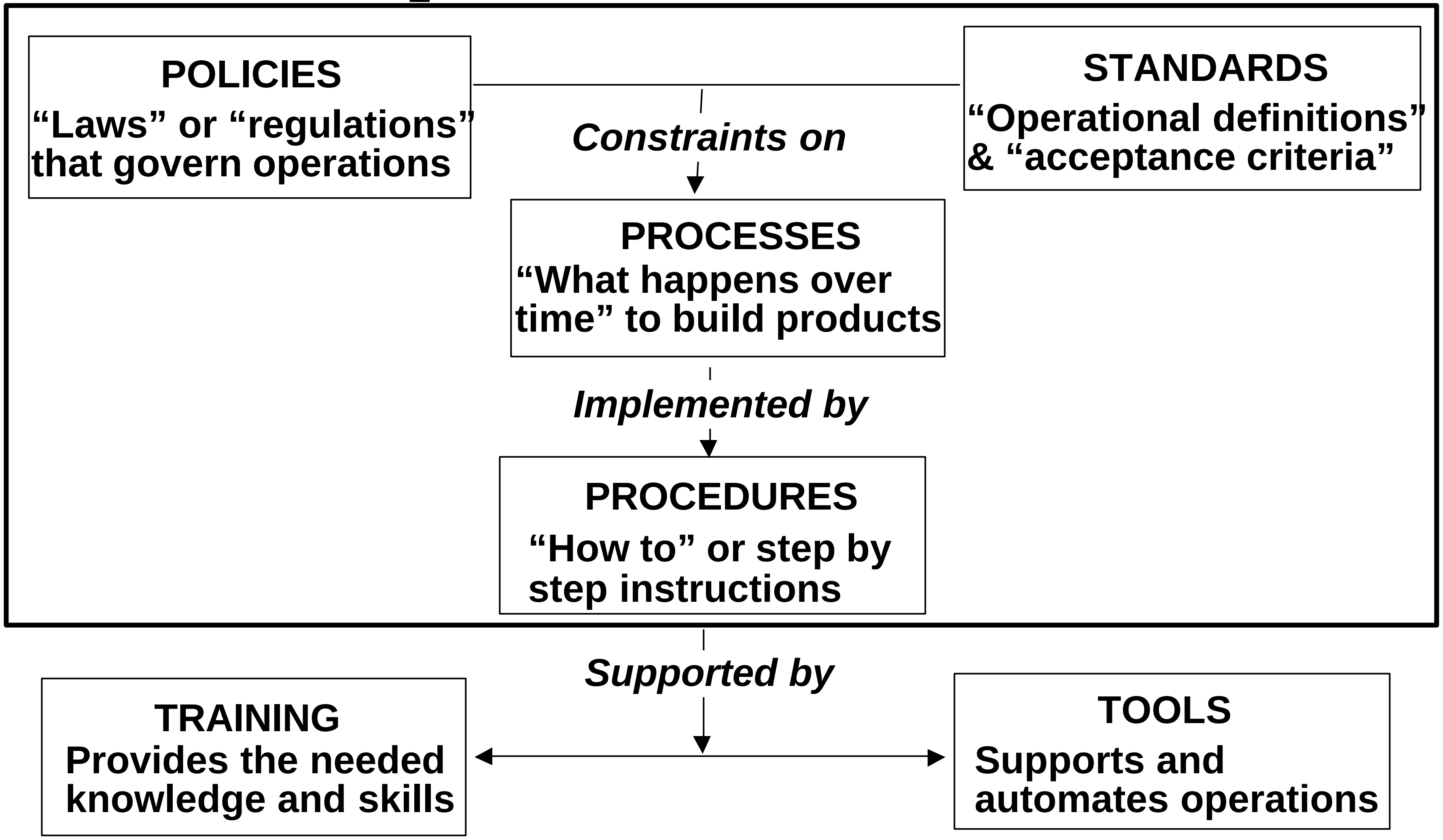


Place the requirements under formal CM and use CCB's to control changes.

3. Example Requirement Metrics

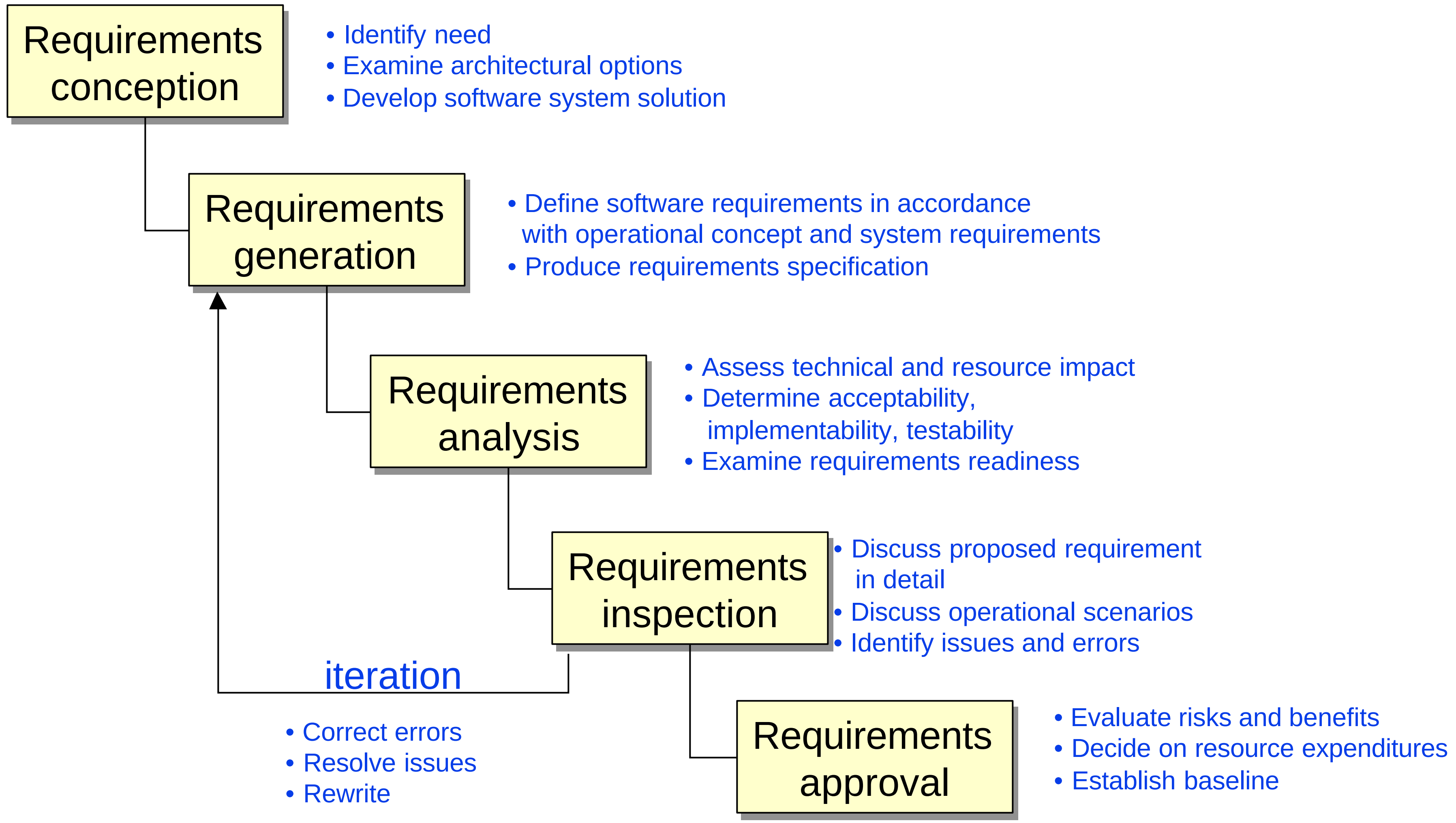
#	Requirement	Reference (e.g., customer)	Allocation	Stability (H/M/L)	Risk (H/M/L)	Priority (H/M/L)
1	System shall send an RTF FAX	SOW # 10-20.3	Software	H	L	M
2	Aircraft position shall be updated by the Inertial Navigation System (INS) Solution	ORD #2-30-20.3.4.4	INS Subsystem Team	M	M	H

4. Operational Framework


POLICIES
"Laws" or "regulations" that govern operations
STANDARDS
"Operational definitions" & "acceptance criteria"
Constraints on
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. Requirement Management Process - 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
iteration
• Correct errors
• Resolve issues
• Rewrite
Requirements approval
• Evaluate risks and benefits
• Decide on resource expenditures
• Establish baseline

4. Manage Requirements Process

Purpose: Effectively Manage Req. Changes

Inputs	Entry	Tasks	eXit	Outputs
• Customer Req. • Product Req. → • Change Requests • Problem Reports	Cust Req. Prod Req. Inspected AND Baselined AND CR/PR is Open	1. PM checks for new CR/PRs 2. Bring CR/PRs to CCB 3. Use CM process **Verification** • Inspection or Peer Review	• CR/PRs are Resolved AND Cust Req. Prod Req. Inspected AND Under CM	• Customer Req. → • Product Req.

Roles: Project Manager (PM), CCB

5. IEEE SyRS/SRS Standard

IEEE System or Software Requirements Standard:

• Reference: "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 830-1993

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 Definitions

#	Requirement (What)	Conditions	Upper Limit	Lower limit	Base Measure
1	System shall send an RTF FAX	user requests	5 seconds	.5 seconds	seconds
2	Aircraft position shall be updated by the Inertial Navigation System (INS) Solution	When INS data is valid	100 milli-seconds	50 milli-seconds	Milli-seconds

Outline

Why Focus on Requirements?

A Practical Requirements Classification

CMMI^{SM} Requirements Overview

Practical Approaches for Requirements

Requirement Examples

Some Advanced Approaches

Summary

Some Advanced Approaches

<u>Juran Model:</u> Customer requirements are written in the customer's language, then translated into the product requirements written in producer's language. This can work with CMMI^{SM}!

<u>QFD/Juran's Quality Planning Process:</u> Measurable requirements that meet customer needs using a defined process.

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

Requirements written in formal languages.

Outline

Summary

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

The top requirements problems in software 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 approaches that you can use today that will help you address problems with requirements.

References

- [Beichter 84] Beichter, F., et al. "SLAN-4-A Software Specification and Design Language." IEEE Transactions on Software Engineering, SE-10, 2 (March 1984), pp. 155-162.
- [Billings 94] Billings, C., et al. "Journey to a Mature Software Process", IBM Systems Journal, vol. 33, no. 1, 1994.
- [Brooks 87] Brooks, Fredrick P., Jr. "No Silver Bullet: Essence and Accidents of Software Engineering". IEEE Computer, 10-19, April 1987.
- [Christel 92] Christel, Michael G. and Kang, Kyo C. "Issues in Requirements Elicitation", CMU/SEI-92-TR-12, 1992.
- [CMMI 02] "Capability Maturity Model® Integration (CMMISM), Version 1.1", CMU/SEI-2002-TR-011, March 2002
- [Ebenau 94] Ebenau, B. and Strauss, S., Software Inspection Process. McGraw-Hill, 1994.
- [Fagan 86] Fagan, M. "Advances in Software Inspections", M. IEEE Transactions on Software Engineering, July 1986
- [Gilb 93] Gilb, T. and Graham, D. Software Inspection. Addison-Wesley, 1993.
- [Humphrey 89] Humphrey, W. S. Managing the Software Process. Reading, MA: Addison-Wesley Publishing Company, 1989.
- [IEEE 93] "IEEE Recommended Practice for Software Requirements Specifications", IEEE Std 830-1993.
- [IEEE 96] "IEEE Guide for Developing System Requirements Specifications", IEEE Std 1233-1996.
- [Jorgensen 87] Jorgensen, Paul C. "Requirements Specification Overview", SEI Curriculum Module SEI-CM-1-1.2 (preliminary), July 1987.
- [Juran 88] Juran, J. and Gryna, F. Juran's Quality Control Handbook. McGraw-Hill, Fourth Edition, 1988.
- [Mays 90] Mays, et al. "Experiences with Defect Prevention". IBM Systems Journal, 1990.
- [McDermid 89] 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, November 1989.
- [McMenamin 84] McMenamin, S. and Palmer, J. Essential Systems Analysis. Yourdon Press Computing Series, Prentice-Hall, 1984.
- [Mettala 92] Mettala, E., and Graham, M. "The Domain-Specific Software Architecture Program", CMU/SEI-92-SR-009.
- [NASA 96] Requirements Capture and Evaluation Process" Training Notebook, Lockheed Martin Space Information Systems (NASA Shuttle Onboard Software - SEI Level 5 Project), 1996.
- [Olson 94] Olson, Timothy G., et al. "A Software Process Framework for the SEI Capability Maturity Model", CMU/SEI-94-HB-01, 1994.
- [Olson 95] Olson, Timothy G. "A Software Quality Strategy for Demonstrating Early ROI", Society of Software Quality Journal, May 1995.
- [Paulk 93] Paulk, Mark C., et al. Capability Maturity Model for Software, Version 1.1 (CMU/SEI-93-TR-24). Pittsburgh, PA: Carnegie Mellon University, 1993.
- [SEI 90] "Software Engineering Institute Affiliates Symposium 1990" proceedings. Pittsburgh, PA: Carnegie Mellon University, 1990.

How to Practically Improve Your Requirements Process Using the CMMI^{SM} Framework

2002 CMMI^{SM} Technology Conference

Tim Olson, President
Quality Improvement Consultants, Inc. (QIC)
Juran Institute Associate
Authorized SEI Lead Assessor
(760) 804-1405
Tim.Olson@att.net

Dennis Beeson
QIC Associate
SEI Candidate Lead Assessor
(760) 375-3376
ddbeeson@gte.net