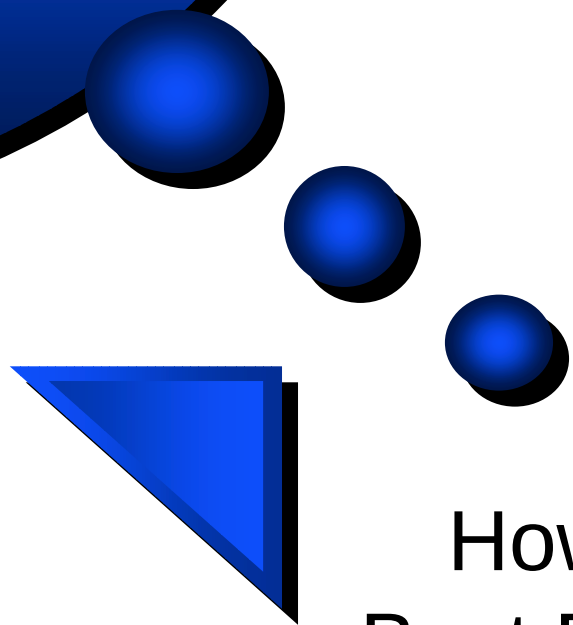


How to Use Standards as Best Practice Information Aids for CMMI[SM]-Compliant Process Engineering

Paul R. Croll

Chair, IEEE Software Engineering Standards Committee

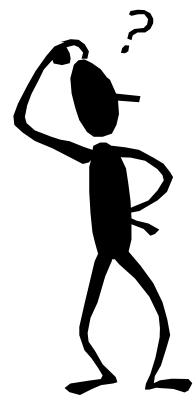
Convener, ISO/IEC JTC1/SC7 WG9

Computer Sciences Corporation

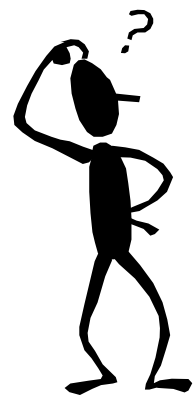
pcroll@csc.com

A Preparatory Exercise

How many of you know that there are two Framework Standards for System and Software Life Cycle Processes?



How many of you know that there are thirty+ Supporting Standards for Software and Systems Process Engineering?



Topics

- Standards As Sources of Best Practice Information
- The CMMI[SM] As A Process Framework Model
- Process Framework Standards
- Best Practice Support for Life Cycle Frameworks
- Applying Best Practices in the CMMI[SM] Context
- Next Steps

Standards As Sources of Best Practice Information

Standards, are consensus-based documents that codify best practice. Consensus-based standards have seven essential attributes that aid in process engineering. They:

- represent the collected experience of others who have been down the same road,
- tell in detail what it means to perform a certain activity,
- can be attached to or referenced by contracts,
- help to assure that two parties have the same meaning for an engineering activity,
- increase professional discipline,
- protect the business and the buyer,
- improve the product.

The CMMI[SM] As A Process Framework Model

- Contains the essential elements of effective processes for one or more disciplines
- Contains a framework that provides the ability to generate multiple models and associated training and assessment materials. These models may represent:
  - software and systems engineering
  - integrated product and process development
  - new disciplines
  - combinations of disciplines
- Provides guidance to use when developing processes

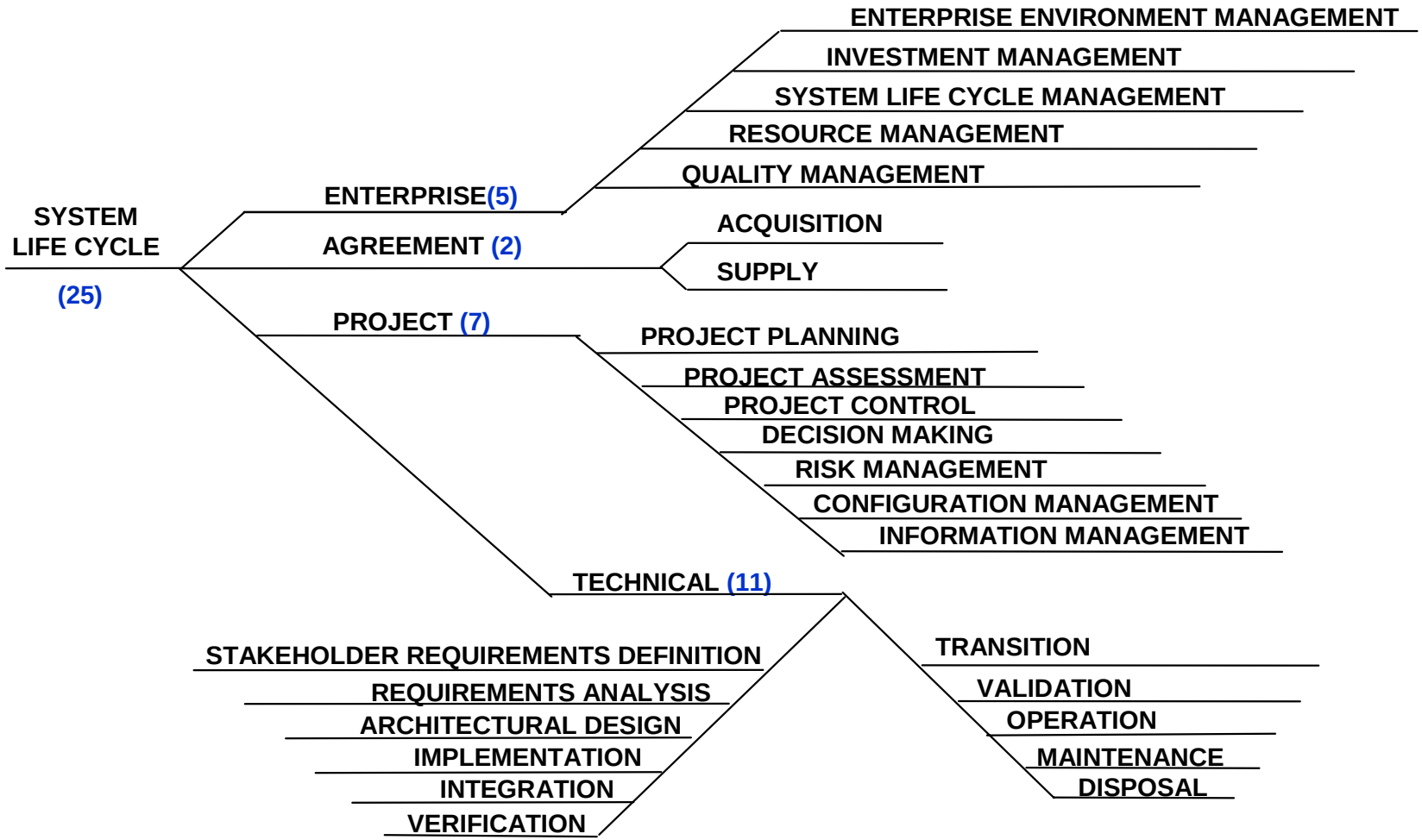
What The CMMI[SM] Is Not

- CMMI models are not processes or process descriptions. Actual processes depend on:
  - Application domain(s)
  - Organization structure
  - Organization size
  - Organization culture
  - Customer requirements or constraints

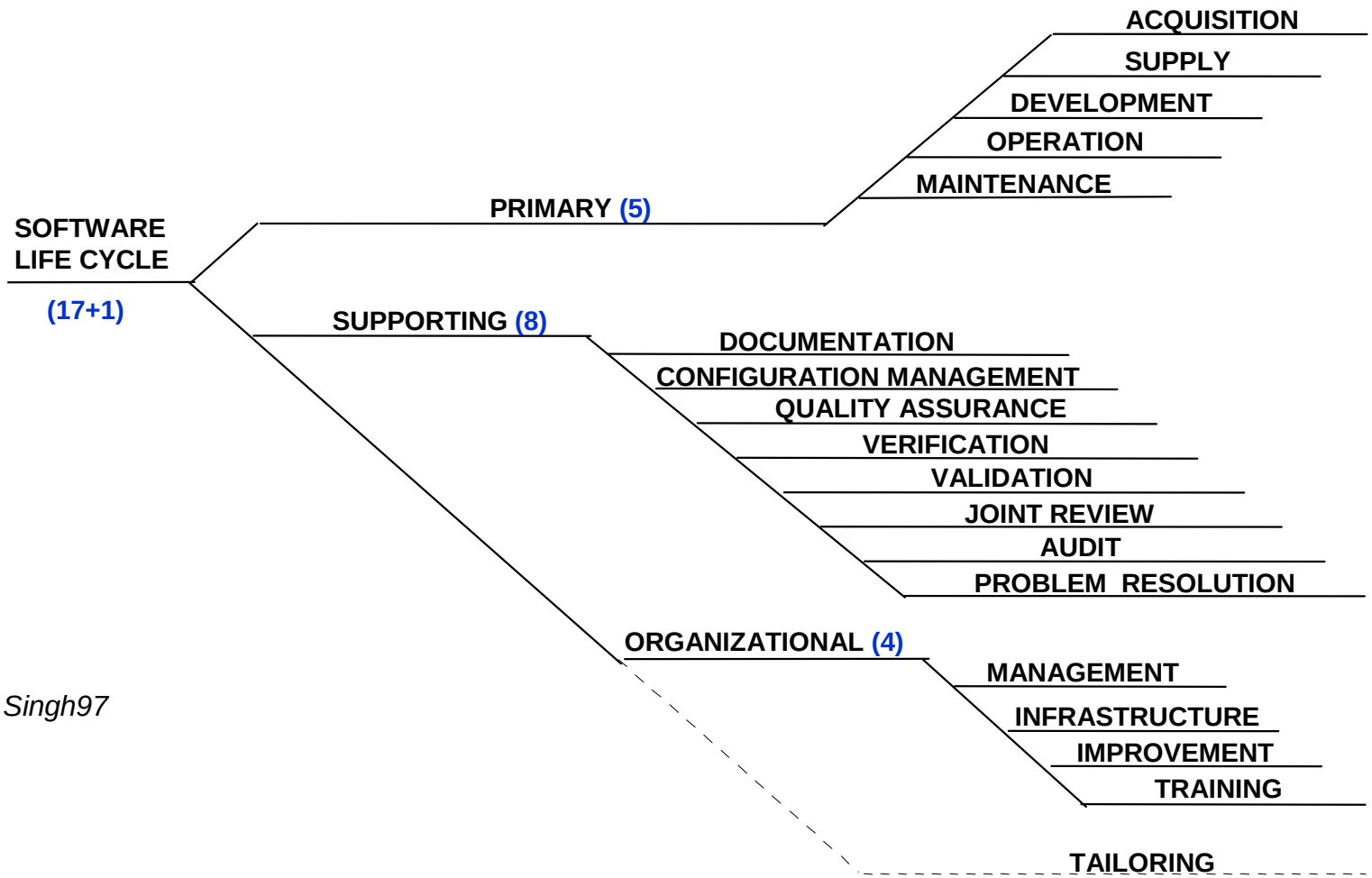
Process Framework Standards

- Systems Life Cycle
  - ISO/IEC 15288
- Software Life Cycle
  - ISO/IEC 12207
  - IEEE/EIA 12207.0, 12207.1, 12207.2
- Process Assessment
  - ISO/IEC 15504

The ISO/IEC 15288 Systems Life Cycle Process Framework

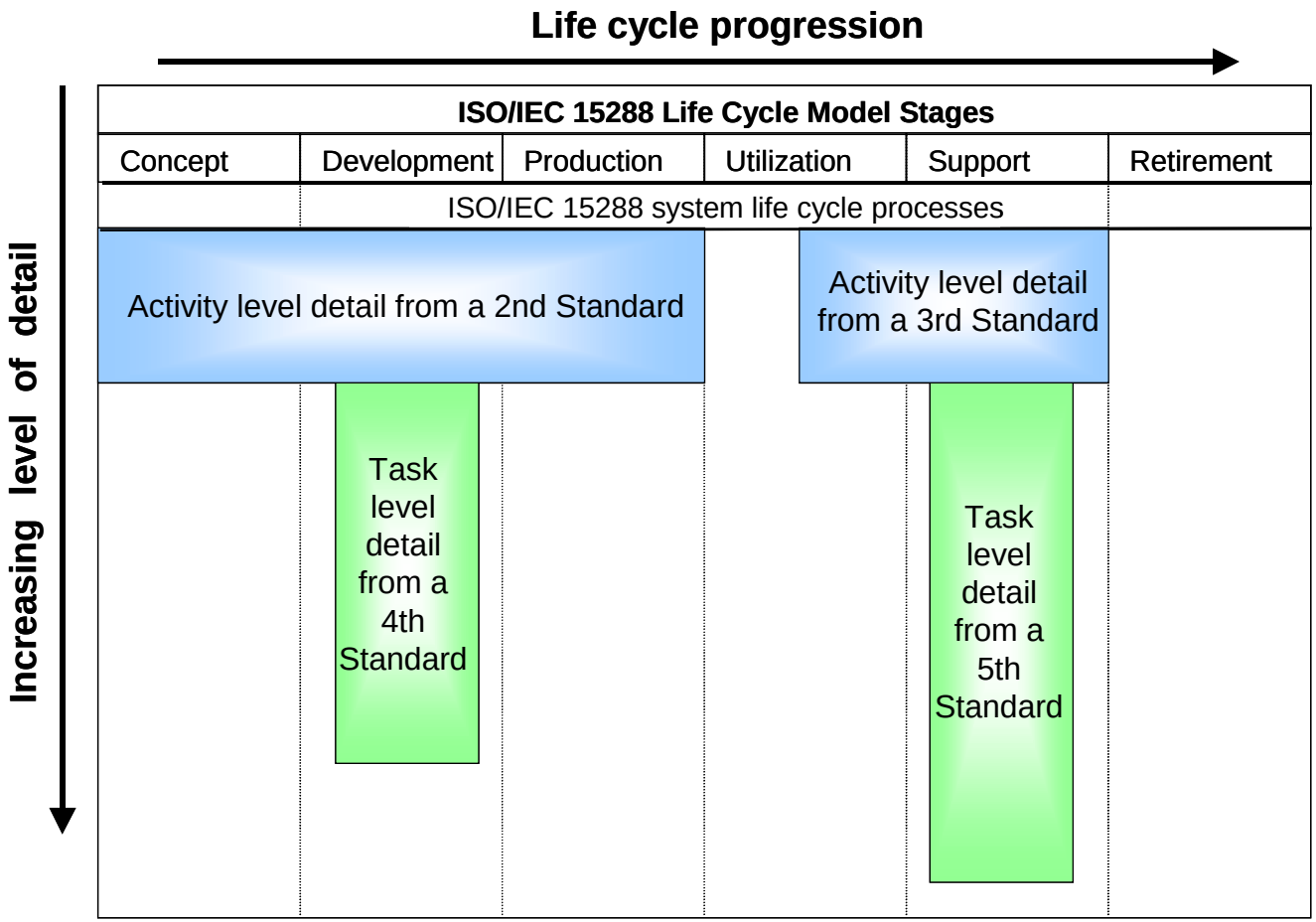


The IEEE/EIA 12207 Software Life Cycle Process Framework



Source: Singh97

Best Practice Support for the System Life Cycle Framework



A1 - ISO/IEC 15288 and other engineering standards

Source: Guide for ISO/IEC 15288 (System Life Cycle Processes), PDTR, © ISO/IEC2002. All rights reserved.

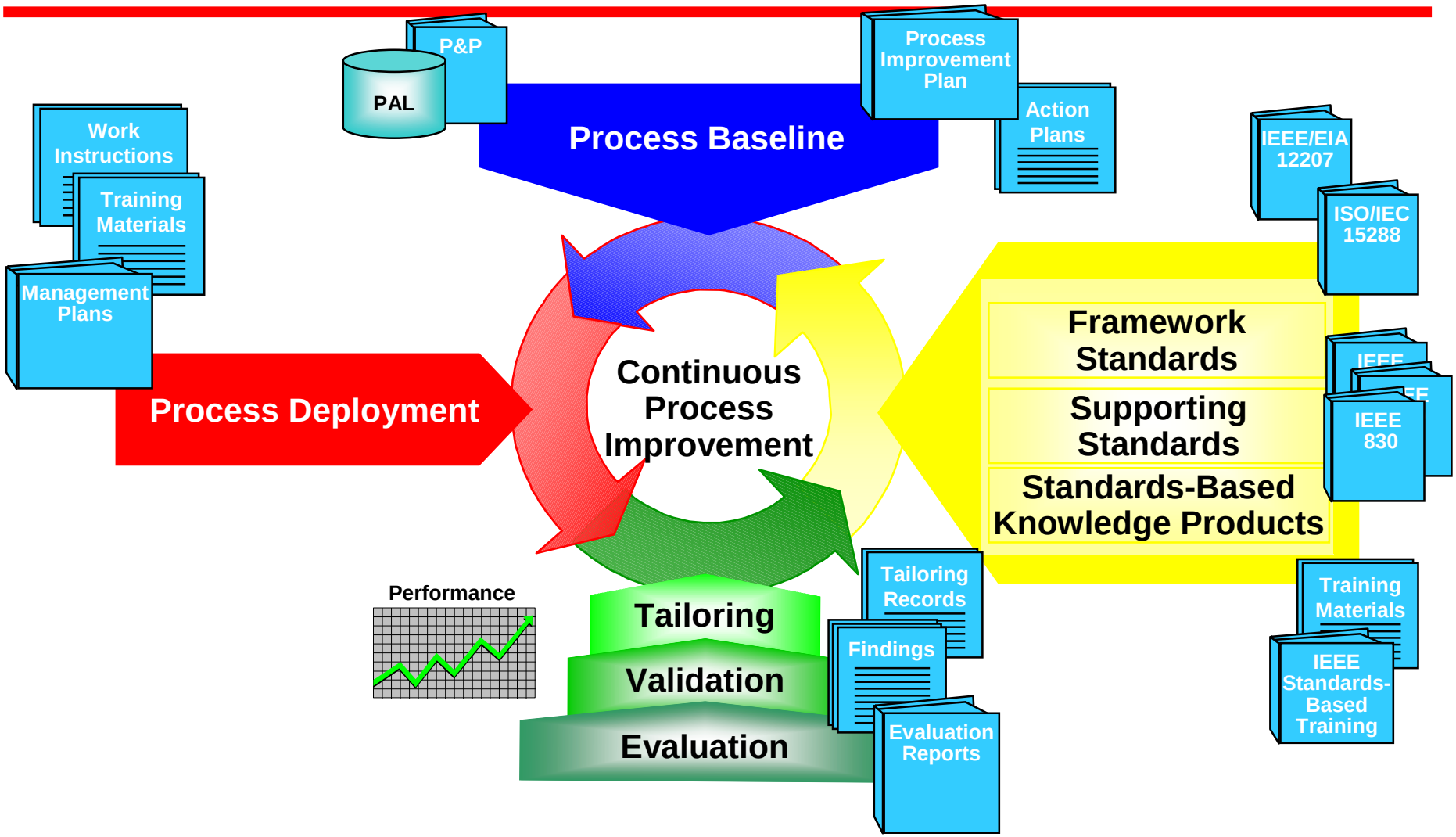
Best Practice Support for the Software Life Cycle Framework

Table 1—Information item matrix

Information item(s)	IEEE/EIA 12207.0 Clause	Kind of documentation	IEEE/EIA 12207.1 Clause	References (See annex A.)
Acceptance strategy and conditions record	5.1.1.9	Record (5.4)	—	IEEE 1062
Acquisition plan	5.1.1.8	Plan	6.1	ASTM E731, E1206, IEEE 1062
Acquisition requirements record	5.1.2.1	Record (5.4)	—	IEEE 1062, 1220
Audit agenda record	6.7.1.4	Record (5.4)	—	—
Audit procedure	6.7.1.4	Procedure (5.3)	—	—
Change request	5.4.4, 5.5.1, 6.2.3	Request	6.2	—
Concept of operations description	5.1.1.1	Description	6.3	IEEE 1362, EIA/IEEE J-STD-016 F.2.1. Also see ISO 5806, 5807, 8631, 8790, and 11411 for guidance on use of notations.
Concept/need determination record	5.1.1.1	Record (5.4)	—	IEEE 1062, 1220
Database design description	5.3.5.3, 5.3.6.3, 5.3.7.1	Description	6.4	IEEE 1016, EIA/IEEE J-STD-016 G.2.3
Detailed design evaluation record	5.3.6.7	Record	6.6	—
Development process plan	5.3.1.4	Plan	6.5	ASTM E622, E1340, EIA/IEEE J-STD-016 E.2.1, IEEE 1074, 1074.1

Source: IEEE/EIA 12207.1-1997, © IEEE 1998. All rights reserved.

Applying Best Practices



CMMI[SM] SE/SW/IPPD/SS v1.1 Process Areas

Process Management

- **Organizational Process Focus**
- **Organizational Process Definition**
- **Organizational Training**
- **Organizational Process Performance**
- **Organizational Innovation and Deployment**

Engineering

- **Requirements Management**
- **Requirements Development**
- **Technical Solution**
- **Product Integration**
- **Verification**
- **Validation**

Project Management

- **Project Planning**
- **Project Monitoring and Control**
- **Supplier Agreement Management**
- **Integrated Project Management for IPPD**
- **Risk Management**
- **Integrated Teaming**
- **Integrated Supplier Management**
- **Quantitative Project Management**

Support

- **Configuration Management**
- **Process and Product Quality Assurance**
- **Measurement and Analysis**
- **Decision Analysis and Resolution**
- **Organizational Environment for Integration**
- **Causal Analysis and Resolution**

CMMI[SM] SE/SW/IPPD/SS v1.1 Mapping - Process Management

Process Management

- EIA 632 - Proc
  Engineering a
- IEEE 1220, Ap
  Management of
  Engineering Process
- IEEE 1
  Life C
- 1517-1

CMMI[SM] SE/SW/IPPD/SS v1.1
Process Area/Specific Practice

- ISO/IEC 15288, System Life Cycle Processes
- IEEE 12207.0, Software Life Cycle
  Processes
- 2207.1, Guide to Software Life
  Processes—Life Cycle Data
- 2207.2, Guide to Software Life
  Cycle Processes—Implementation
  Considerations

Framework Standards

Supporting Standards

CMMI[SM] SE/SW/IPPD/SS v1.1 Mapping - Process Management

Process Management

- EIA 632 - Processes for Engineering a System
- IEEE 1220, Application and Management of the Systems Engineering Process
- IEEE 1074, Developing Software Life Cycle Processes
- 1517-1999, Reuse Processes

- ISO/IEC 15288, System Life Cycle Processes
- IEEE 12207.0, Software Life Cycle Processes
- IEEE 12207.1, Guide to Software Life Cycle Processes—Life Cycle Data
- IEEE 12207.2, Guide to Software Life Cycle Processes—Implementation Considerations

CMMI[SM] SE/SW/IPPD/SS v1.1 Mapping - Project Management

Project Management

- IEEE 1220, Application and Management of the Systems Engineering Process
- IEEE 1058, Software Project Management Plans
- IEEE 1490, A Guide to the Program Management Body of Knowledge
- IEEE 1062, Recommended Practice for Software Acquisition
- IEEE 1540, Risk Management
- IEEE 1028, Software Reviews

- ISO/IEC 15288, System Life Cycle Processes
- IEEE 12207.0, Software Life Cycle Processes
- IEEE 12207.1, Guide to Software Life Cycle Processes—Life Cycle Data
- IEEE 12207.2, Guide to Software Life Cycle Processes—Implementation Considerations

CMMI[SM] SE/SW/IPPD/SS v1.1 Mapping - Engineering

Engineering

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 1362, Guide for Concept of Operations Document
- IEEE 1471, Architectural Description of Software Intensive Systems
- IEEE 830, Software Requirements Specifications
- IEEE 1016, Software Design Descriptions
- IEEE 1012, Software Verification and Validation
- IEEE 1008, Software Unit Testing

- ISO/IEC 15288, System Life Cycle Processes
- IEEE 12207.0, Software Life Cycle Processes
- IEEE 12207.1, Guide to Software Life Cycle Processes—Life Cycle Data
- IEEE 12207.2, Guide to Software Life Cycle Processes—Implementation Considerations

- IEEE 1228, Software Safety Plans
- IEEE 1063, Software User Documentation
- IEEE 1219, Software Maintenance
- IEEE 1320.1,.2, IDEF0, IDEF1X97
- IEEE 1420.1, Data Model for Reuse Library Interoperability

CMMI[SM] SE/SW/IPPD/SS v1.1 Mapping - Support

Support

- IEEE 828, Software Configuration Management Plans
- IEEE 730, Software Quality Assurance Plans
- IEEE 982.1, Dictionary of Measures to Produce Reliable Software
- IEEE 1045, Software Productivity Metrics
- IEEE 1061, Software Quality Metrics Methodology
- IEEE 1219, Software Maintenance

- ISO/IEC 15288, System Life Cycle Processes
- IEEE 12207.0, Software Life Cycle Processes
- IEEE 12207.1, Guide to Software Life Cycle Processes—Life Cycle Data
- IEEE 12207.2, Guide to Software Life Cycle Processes—Implementation Considerations

- IEEE 1465 (ISO/IEC 12119) - Software Packages - Quality Requirements and Testing
- IEEE 14143.1 (ISO/IEC 1443-1) - Functional Size Measurement - Part 1: Definition of Concepts

Risk Management

SP 1.1-1 Determine Risk Sources and Categories

- Determine risk sources and categories

SP 1.2-1 Define Risk Parameters

- Define the parameters used to analyze and classify risks, and the parameters used to control the risk management effort

SP 1.3-1 Establish a Risk Management Strategy

- Establish and maintain the strategy and methods to be used for risk management

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.4.6 - Risk Management Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Annex G - Life Cycle Process Objectives
    - Clause G.10 - Management Process
- IEEE 1540, Risk Management

Risk Management

SP 1.1-1 Determine Risk Sources and Categories

- Determine risk sources a

5.4.6.3 Risk Management Process Activities

The project shall implement the following activities in accordance with applicable organization policies and procedures with respect to the Risk Management Process.

a) Establish a systematic approach to risk identification, determining what could go wrong and would adversely affect the system and/or the organization.

b) Define the risks in terms of their dimensions, e.g. technical, human, cost, schedule.

c) Determine the likelihood of risk occurrence.

d) Evaluate the impact of the risks and define their possible adverse consequences.

e) Prioritize the risks in terms of their likelihood and consequence.

. . .

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.4.6 - Risk Management
    rocess
  - /EIA 12207.0, Software
    Cycle Processes
  - Annex G - Life Cycle Process Objectives
    - Clause G.10 - Management Process
  - 1540, Risk Management

Source:
ISO/IEC CD 15288 FDIS,
© ISO/IEC2002. All rights reserved.

Risk Management

k) Determine the scope of risk management to be performed for the project;
l) Identify risks to the project as they develop;
m) Analyze risks and determine the priority in which to apply resources to mitigate those risks;
n) Define, implement, and assess appropriate risk mitigation strategies;
o) Define, apply, and assess risk metrics to measure the change in the risk state and the progress of the mitigation activities;

SP
an
- ...

SP
- ... analyze and classify risks, and the parameters used to control the risk management effort

SP 1.3-1 Establish a Risk Management Strategy
- Establish and maintain the strategy and methods to be used for risk management

...5288, System Life
...cesses
- ...5.4.6 - Risk Management
...s
- ... 12207.0, Software
...le Processes
  - Annex G - Life Cycle Process Objectives
    - Clause G.10 - Management Process
- IEEE 1540, Risk Management

Source: IEEE/EIA 12207.0-1997, © IEEE 2001. All rights reserved.

5.1.1.2 Establish the risk management process.
A description of the risk management process to be implemented shall be documented and promulgated. The description of the procedures that implement the risk management process should include:
a) The frequency at which risks are to be reanalyzed and monitored
b) The type of risk analysis required (quantitative and/or qualitative)
c) The scales to be used to estimate risk likelihood and consequences and their descriptive and measurement uncertainty
d) The types of risk thresholds to be used
e) The types of measures used to track and monitor the state of the risks
f) How risks are to be prioritized for treatment
g) Which stakeholder(s)perspectives the risk management process supports
h) The risk categories to be considered

D/IEC 15288, System Life
cle Processes

- Clause 5.4.6 - Risk Management Process

EE/EIA 12207.0, Software
e Cycle Processes

- Annex G - Life Cycle Process Objectives
  - Clause G.10 - Management Process

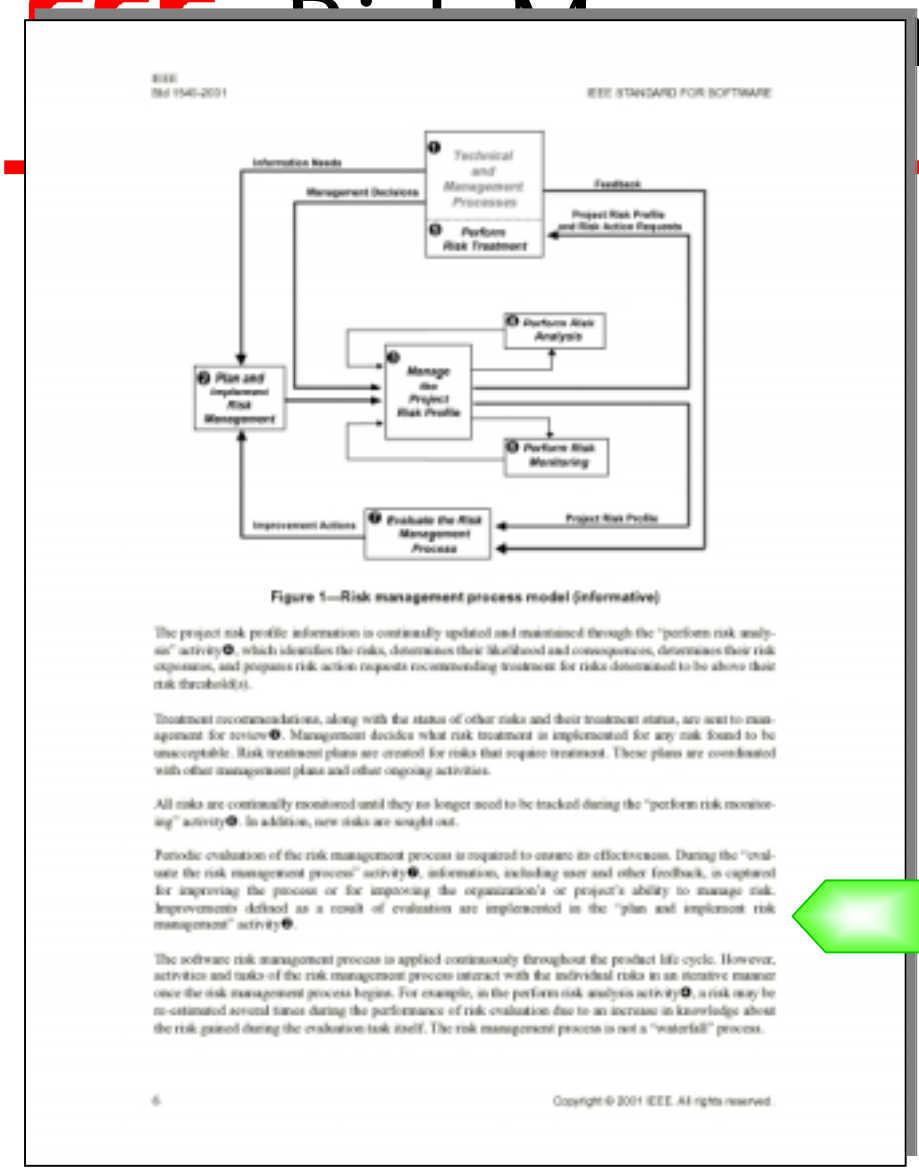
- IEEE 1540, Risk Management

SP 1.3-1 Establish a Risk Management Strategy

- Establish and maintain the strategy and methods to be used for risk management

Source:
IEEE 1540-2000, © IEEE 2001. All rights reserved.

Risk Management



IEEE Std 1540-2001 IEEE STANDARD FOR SOFTWARE

Figure 1—Risk management process model (informative)

The project risk profile information is continually updated and maintained through the "perform risk analysis" activity❷, which identifies the risks, determines their likelihood and consequences, determines their risk exposures, and prepares risk action requests recommending treatment for risks determined to be above their risk threshold(s).

Treatment recommendations, along with the status of other risks and their treatment status, are sent to management for review❸. Management decides what risk treatment is implemented for any risk found to be unacceptable. Risk treatment plans are created for risks that require treatment. These plans are coordinated with other management plans and other ongoing activities.

All risks are continually monitored until they no longer need to be tracked during the "perform risk monitoring" activity❹. In addition, new risks are sought out.

Periodic evaluation of the risk management process is required to ensure its effectiveness. During the "evaluate the risk management process" activity❺, information, including user and other feedback, is captured for improving the process or for improving the organization's or project's ability to manage risk. Improvements defined as a result of evaluation are implemented in the "plan and implement risk management" activity❶.

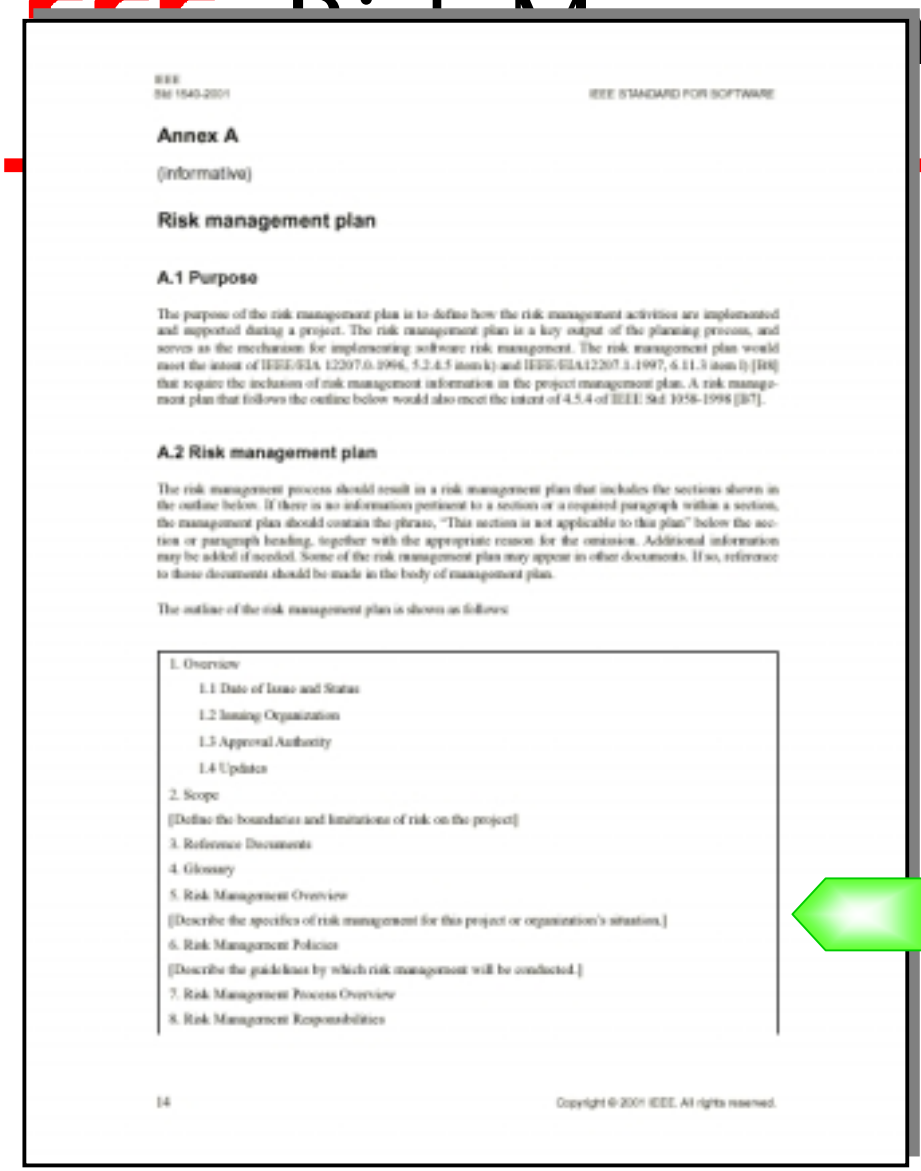
The software risk management process is applied continuously throughout the product life cycle. However, activities and tasks of the risk management process interact with the individual risks in an iterative manner once the risk management process begins. For example, in the perform risk analysis activity❷, a risk may be re-estimated several times during the performance of risk evaluation due to an increase in knowledge about the risk gained during the evaluation task itself. The risk management process is not a "waterfall" process.

6

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.4.6 - Risk Management Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Annex G - Life Cycle Process Objectives
    - Clause G.10 - Management Process
- IEEE 1540, Risk Management

Source:
IEEE 1540-2000, © IEEE 2001. All rights reserved.

Risk Management



IEEE Std 1540-2001 IEEE STANDARD FOR SOFTWARE

Annex A

(informative)

Risk management plan

A.1 Purpose

The purpose of the risk management plan is to define how the risk management activities are implemented and supported during a project. The risk management plan is a key output of the planning process, and serves as the mechanism for implementing software risk management. The risk management plan would meet the intent of IEEE/EIA 12207.0-1996, 5.2.4.5 item k) and IEEE/EIA12207.1-1997, 6.11.3 item l) [B8] that require the inclusion of risk management information in the project management plan. A risk management plan that follows the outline below would also meet the intent of 4.5.4 of IEEE Std 1058-1998 [B7].

A.2 Risk management plan

The risk management process should result in a risk management plan that includes the sections shown in the outline below. If there is no information pertinent to a section or a required paragraph within a section, the management plan should contain the phrase, "This section is not applicable to this plan" below the section or paragraph heading, together with the appropriate reason for the omission. Additional information may be added if needed. Some of the risk management plan may appear in other documents. If so, reference to those documents should be made in the body of management plan.

The outline of the risk management plan is shown as follows:

1. Overview
   - 1.1 Date of Issue and Status
   - 1.2 Issuing Organization
   - 1.3 Approval Authority
   - 1.4 Updates
2. Scope

[Define the boundaries and limitations of risk on the project]

3. Reference Documents
4. Glossary
5. Risk Management Overview

[Describe the specifics of risk management for this project or organization's situation.]

6. Risk Management Policies

[Describe the guidelines by which risk management will be conducted.]

7. Risk Management Process Overview
8. Risk Management Responsibilities

14 Copyright © 2001 IEEE. All rights reserved.

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.4.6 - Risk Management Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Annex G - Life Cycle Process Objectives
    - Clause G.10 - Management Process
- IEEE 1540, Risk Management

Source:
IEEE 1540-2000, © IEEE 2001. All rights reserved.

Requirements Development

SP 2.1-1 Establish Product and Product Component Requirements

- Establish and maintain, from the customer requirements, product and product component requirements essential to product and product component effectiveness and affordability

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.5.3 - Requirements Analysis Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Clause 5.3.2 - System Requirements Analysis
  - Clause 5.3.4 - Software requirements analysis

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 830, Software Requirements Specifications

Requirements Development

SP 2.1-1 Establish Product and Product Component Requirements

- Establish and maintain, from th

5.5.3 Requirements Analysis Process

5.5.3.1 Purpose of the Requirements Analysis Process

The purpose of the Requirements Analysis Process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services. This process builds a representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measurable system requirements that specify, from the developer's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements.

5.5.3.2 Requirements Analysis Process Outcomes

As a result of the successful implementation of the Requirements Analysis Process:

a) The required characteristics, attributes, and functional and performance requirements for a product solution are specified.

b) Constraints that will affect the architectural design of a system and the means to realize it are specified.

c) The integrity and traceability of system requirements to stakeholder requirements is achieved**. . .**

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.5.3 - Requirements Analysis rocess
  - A 12207.0, Software Life Cycle s
  - se 5.3.2 - System Requirements ysis
  - se 5.3.4 - Software requirements sis

33, Guide for Developing System nents Specifications

0, Software Requirements tions

Source:
ISO/IEC CD 15288 FDIS, © ISO/IEC2002.
All rights reserved.

evelopment

5.3.2.1 The specific intended use of the system to be developed shall be analyzed to specify system requirements. The system requirements specification shall describe: functions and capabilities of the system; business, organizational and user requirements; safety, security, human-factors engineering (ergonomics), interface, operations, and maintenance requirements; design constraints and qualification requirements. The system requirements specification shall be documented.
5.3.4.1 The developer shall establish and document software requirements, including the quality characteristics specifications, described below. . . .
a) Functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the software item is to perform;
b) Interfaces external to the software item;
c) Qualification requirements;
d) Safety specifications, including those related to methods of operation and maintenance, environmental influences, and personnel injury;
e) Security specifications, including those related to compromise of sensitive information . . .

SO/IEC 15288, System Life Cycle rocesses

- Clause 5.5.3 - Requirements Analysis Process

EE/EIA 12207.0, Software Life Cycle ocesses

- Clause 5.3.2 - System Requirements Analysis
- Clause 5.3.4 - Software requirements analysis

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 830, Software Requirements Specifications

Source:
IEEE/EIA 12207.0-1997,

elopment

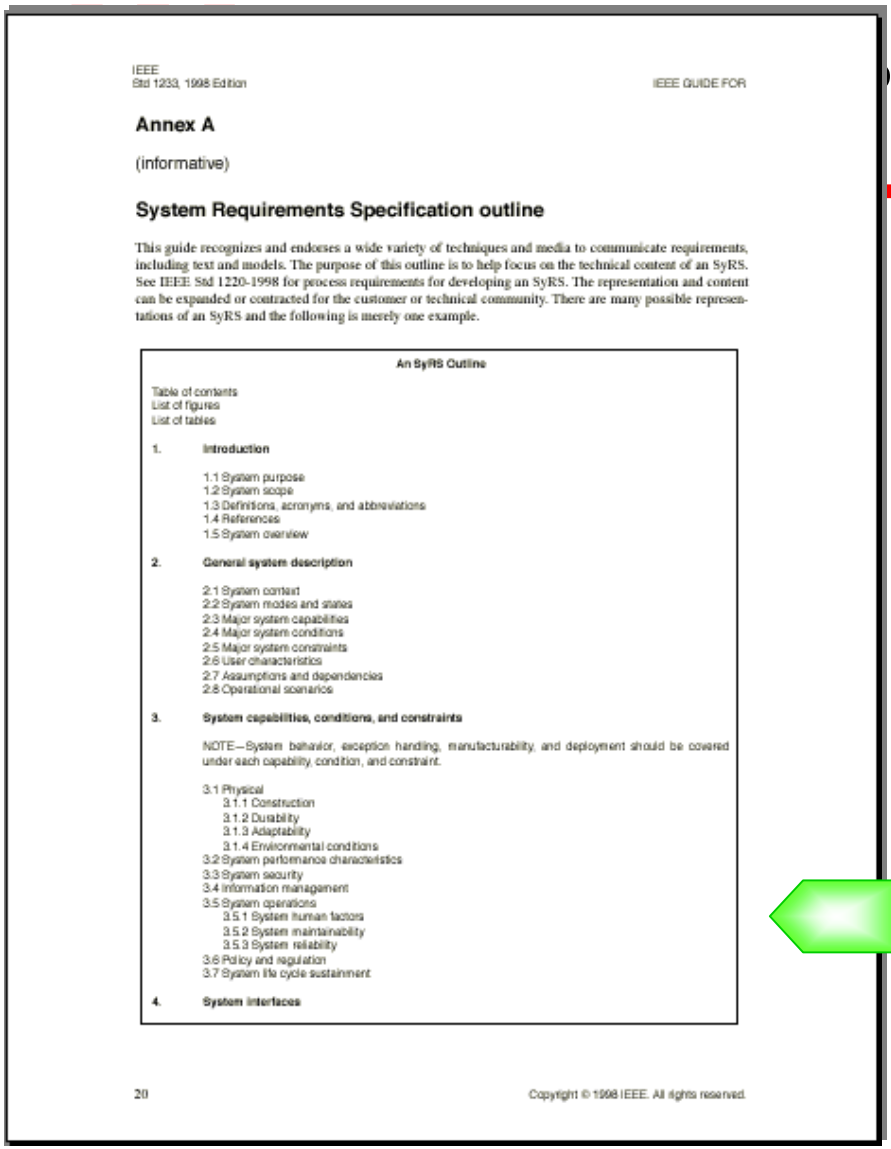
7.2 Build a well-formed requirement
The analysts carry out this subphase by doing the following:
a) Ensuring that each requirement is a necessary, short, definitive statement of need (capability, constraints);
b) Defining the appropriate conditions (quantitative or qualitative measures) for each requirement and avoiding adjectives such as "resistant" or "industry wide;"
c) Avoiding requirements pitfalls (see 6.4);
d) Ensuring the readability of requirements, which entails the following:
1) Simple words/phrases/concepts;
2) Uniform arrangement and relationship;
3) Definition of unique words, symbols, and notations;
4) The use of grammatically correct language and symbology.
e) Ensuring testability.
Example:
Capability: Move people between Los Angeles and New York
Condition: Cruising speed of 200 km/hr
Constraint: Maximum speed of 300 km/hr
Well-formed requirement: This system should move people between Los Angeles and New York at an optimal cruising speed of 200 km/hr with a maximum speed of 300 km/hr.

Source:
IEEE 1233-1998, © IEEE 1998. All rights reserved.

EC 15288, System Life Cycle
sses
lause 5.5.3 - Requirements Analysis
rocess
/EIA 12207.0, Software Life Cycle
sses
lause 5.3.2 - System Requirements
nalysis
lause 5.3.4 - Software requirements
nalysis

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 830, Software Requirements Specifications

Development



IEEE
Std 1233, 1998 Edition

IEEE GUIDE FOR

Annex A

(informative)

System Requirements Specification outline

This guide recognizes and endorses a wide variety of techniques and media to communicate requirements, including text and models. The purpose of this outline is to help focus on the technical content of an SyRS. See IEEE Std 1220-1998 for process requirements for developing an SyRS. The representation and content can be expanded or contracted for the customer or technical community. There are many possible representations of an SyRS and the following is merely one example.

An SyRS Outline

Table of contents
List of figures
List of tables

1. **Introduction**
   - 1.1 System purpose
   - 1.2 System scope
   - 1.3 Definitions, acronyms, and abbreviations
   - 1.4 References
   - 1.5 System overview
2. **General system description**
   - 2.1 System context
   - 2.2 System modes and states
   - 2.3 Major system capabilities
   - 2.4 Major system conditions
   - 2.5 Major system constraints
   - 2.6 User characteristics
   - 2.7 Assumptions and dependencies
   - 2.8 Operational scenarios
3. **System capabilities, conditions, and constraints**

   NOTE—System behavior, exception handling, manufacturability, and deployment should be covered under each capability, condition, and constraint.

   - 3.1 Physical
     - 3.1.1 Construction
     - 3.1.2 Durability
     - 3.1.3 Adaptability
     - 3.1.4 Environmental conditions
   - 3.2 System performance characteristics
   - 3.3 System security
   - 3.4 Information management
   - 3.5 System operations
     - 3.5.1 System human factors
     - 3.5.2 System maintainability
     - 3.5.3 System reliability
   - 3.6 Policy and regulation
   - 3.7 System life cycle sustainment
4. **System interfaces**

20

- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.5.3 - Requirements Analysis Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Clause 5.3.2 - System Requirements Analysis
  - Clause 5.3.4 - Software requirements analysis

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 830, Software Requirements Specifications

Source:
IEEE 1233-1998, © IEEE 1998. All rights reserved.

elopment

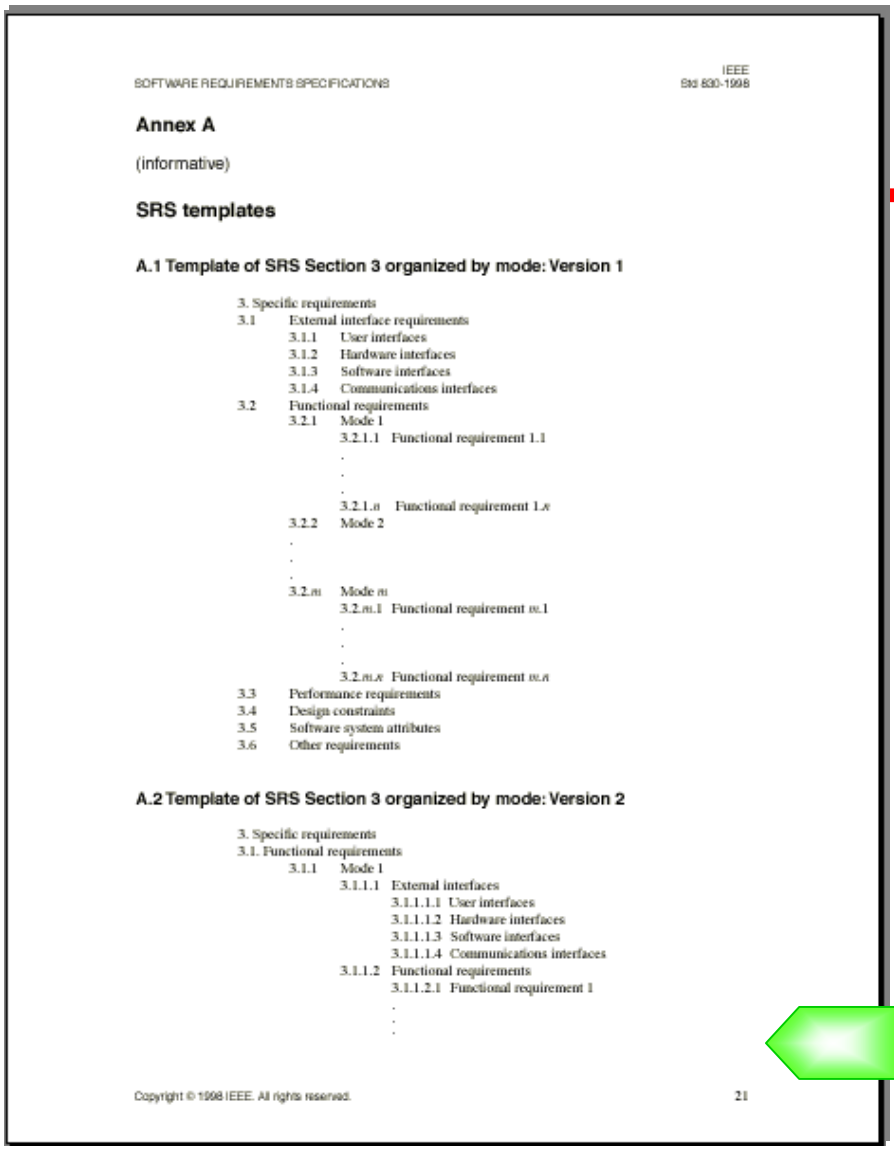
5.3.2 Functions
Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall”
These include:
a) Validity checks on the inputs
b) Exact sequence of operations
c) Responses to abnormal situations, including:
  1) Overflow
  2) Communication facilities
  3) Error handling and recovery
d) Effect of parameters
e) Relationship of outputs to inputs . . .
  1) It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does
not imply that the software design will also be partitioned that way.
5.3.3 Performance requirements
This subsection should specify both the static and the dynamic numerical requirements placed on the soft-
ware or on human interaction with the software as a whole. Static numerical requirements may include the
following:
a) The number of terminals to be supported;
b) The number of simultaneous users to be supported;
c) Amount and type of information to be handled.

Source:
IEEE 830-1998, © IEEE 1998. All rights reserved.

EC 15288, System Life Cycle
sses
lause 5.5.3 - Requirements Analysis
rocess
/EIA 12207.0, Software Life Cycle
sses
lause 5.3.2 - System Requirements
nalysis
lause 5.3.4 - Software requirements
nalysis

E 1233, Guide for Developing System
equirements Specifications
IEEE 830, Software Requirements Specifications

Development



SOFTWARE REQUIREMENTS SPECIFICATIONS IEEE Std 830-1998

Annex A

(informative)

SRS templates

A.1 Template of SRS Section 3 organized by mode: Version 1

3. Specific requirements
3.1 External interface requirements
3.1.1 User interfaces
3.1.2 Hardware interfaces
3.1.3 Software interfaces
3.1.4 Communications interfaces
3.2 Functional requirements
3.2.1 Mode 1
3.2.1.1 Functional requirement 1.1
.
.
.
3.2.1.*n* Functional requirement 1.*n*
3.2.2 Mode 2
.
.
.
3.2.*m* Mode *m*
3.2.*m*.1 Functional requirement *m*.1
.
.
.
3.2.*m*.*n* Functional requirement *m*.*n*
3.3 Performance requirements
3.4 Design constraints
3.5 Software system attributes
3.6 Other requirements

A.2 Template of SRS Section 3 organized by mode: Version 2

3. Specific requirements
3.1. Functional requirements
3.1.1 Mode 1
3.1.1.1 External interfaces
3.1.1.1.1 User interfaces
3.1.1.1.2 Hardware interfaces
3.1.1.1.3 Software interfaces
3.1.1.1.4 Communications interfaces
3.1.1.2 Functional requirements
3.1.1.2.1 Functional requirement 1
.
.
.

Copyright © 1998 IEEE. All rights reserved. 21

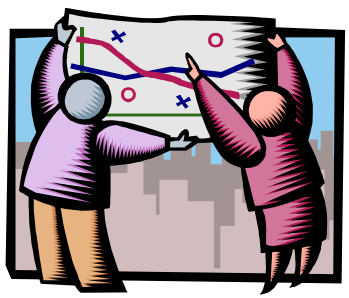
- ISO/IEC 15288, System Life Cycle Processes
  - Clause 5.5.3 - Requirements Analysis Process
- IEEE/EIA 12207.0, Software Life Cycle Processes
  - Clause 5.3.2 - System Requirements Analysis
  - Clause 5.3.4 - Software requirements analysis

- IEEE 1233, Guide for Developing System Requirements Specifications
- IEEE 830, Software Requirements Specifications

Source:
IEEE 830-1998, © IEEE 1998. All rights reserved.

8 Steps to Success With Best Practice Information Aids

1. Understand your business processes

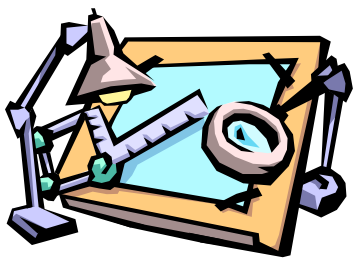


2. Look to the CMMI[SM] for Process Completeness

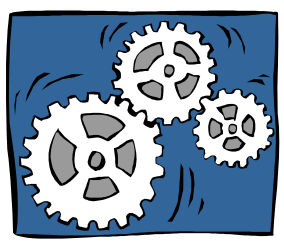
3. Look to Framework Standards for Life Cycle Definition

4. Look to Supporting Standards for Process Detail

5. Build Your Process Architecture



6. Execute Your Processes



7. Measure Your Results - Modify Processes as Necessary

8. Confirm Your Status With Independent Appraisals

Next Steps

- The IEEE Software Engineering Standards Collection and the CMMI[SM]
  - Mapping Standards to the CMMI[SM]
  - Aligning the Collection with the CMMI[SM]
  - Updating IEEE Standards-Based Training
  - Establishing web-based assets
- ISO/IEC Framework Standards and the CMMI[SM]
  - Harmonizing the CMMI[SM] with the Systems and Software Life Cycle Framework Standards

How you can help . . .

- Individuals
  - Participate with as a SESC Working Group or Study Group member
- Organizations
  - Participate as part of the Stakeholders Advisory Group
- Tool vendors
  - Collaborate on implementing standards-based practices in your tool sets
- Software radicals/paradigm shifters
  - Help us understand issues like “product assurance in an agile programming environment”

For more information . . .

Paul R. Croll
Computer Sciences Corporation
5166 Potomac Drive
King George, VA 22485-5824

Phone: +1 540.663.9251
Fax: +1 540.663.0276
e-mail: pcroll@csc.com

For IEEE Standards:

http://computer.org/standards/sesc/
http://computer.org/cspress/CATALOG/st01110.htm

For ISO/IEC Standards:

http://saturne.info.uqam.ca/Labo_Recherche/Lrgl/sc7/

Questions?

References

CMMI[SM] -SE/SW/IPPD/SS, V1.1, *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, and Supplier Sourcing Version 1.1, CMMI[SM] -SE/SW/IPPD/SS, V1.1, Continuous Representation.* CMU/SEI-CMU/SEI-2002-TR-011, ESC-TR-2002-011, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, March 2002.

Guide for ISO/IEC 15288 (System Life Cycle Processes), PDTR, ISO/IEC JTC1/SC7, 2002.

IEEE Standard 830-1998, *Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers, Inc. New York, NY, 1998.

References - 2

IEEE Standard 1233-1998, *Guide for Developing System Requirements Specifications*, Institute of Electrical and Electronics Engineers, Inc. New York, NY, 1998.

IEEE Standard 1540-2001, *IEEE Standard for Software Life Cycle Processes — Risk Management*, Institute of Electrical and Electronics Engineers, Inc. New York, NY, 2001.

IEEE/EIA Standard 12207.0-1996, *Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes*, Institute of Electrical and Electronics Engineers, Inc. New York, NY, 1998.

References - 3

IEEE/EIA Standard 12207.1-1997, *Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes - Implementation Considerations*, Institute of Electrical and Electronics Engineers, Inc. New York, NY, 1998.

ISO/IEC CD 15288 FDIS:2002, *Systems Engineering — System Life Cycle Processes*, ISO/IEC JTC1/SC7, 2002.

[Singh97] Raghu Singh, *An Introduction to International Standards ISO/IEC 12207, Software Life Cycle Processes*, 1997.