

# The Software Development Standards

Instructor: Dr. Hany H. Ammar  
Dept. of Computer Science and  
Electrical Engineering, WVU

---



# OUTLINE

- The System Life Cycle Model and the System Development Process.
- Software Engineering and the Software Development Process
- **Software Development standards**
- **The ICASE Environments**
- **ICASE Tool: Teamwork (see notes of Ch. 2), and Software through Pictures (StP)**

# Software Development Standards

Three standards for software development are discussed

- The software engineering standard (PSS-05-0) of the European Space Agency (ESA )
- The MIL-STD-498 standard for software development of the US Department of Defense
- IEEE/EIA 12207 Standard for Information Technology-software life cycle processes



# The ESA standard

- The ESA standard consists of two parts namely, the *product standards* part and the *procedure standards* part.
- The product standards part contains standards, recommendations and guidelines concerning the software product,
- The procedure standards part describes the procedures used to manage the software development project.



# The ESA standard

The ESA product standard mandates that all software projects shall have a life cycle approach consisting of the following basic phases:

- User Requirements (UR) definition
- Software Requirements (SR) specification
- Architectural Design (AD) specification
- Detailed Design (DD) and production of the code
- Transfer (TR) of software to operation, and
- Operations and Maintenance (OM).



# The ESA standard

- The *UR phase* is the problem definition phase in which the scope of the software is clearly specified by the users in cooperation with the developer's teams of software engineers, hardware engineers, and managers.
- In the *UR phase*, the operational environment of the software is determined.
- The users requirements are captured and documented in a *User Requirements Doc (URD)*.
- The review of the *URD (UR/R)* is performed by the same teams who have already worked on software specifications in the UR phase.

# The ESA standard

- The *SR phase* is the software requirements analysis and specification phase
- A **logical model** of the software is produced (using Structured analysis or Object-Oriented Analysis) and used to analyze the completeness, consistency, and testability of the requirements
- *Software Requirements Document (SRD)* is produced and is formally reviewed (SR/R) by the users, software engineers, hardware engineers, and managers concerned.

# The ESA standard

- The *Architecture Design (AD)* phase deals with the construction of what is termed as “the **physical model**” of the software. It defines the architecture or structure of the software in terms of components or modules and their interfaces
- The software components as well as the data flow and control flow between them are defined.
- The deliverable produced in this phase is the *Architectural Design Document (ADD)*. The ADD is again formally reviewed (AD/R) by the same teams mentioned above.



# The ESA standard

- The activities of the *DD phase* include module design, coding, unit testing, integration testing and system testing.
- A *Detailed Design Document (DDD)* and the *Software User Manual (SUM)* are produced concurrently with coding and testing
- Unit, integration, and system testing is performed according to verification plans established in the SR and AD

# The ESA standard

- The code, DDD, and SUM documents are reviewed in the *formal Detailed Design Review (DD/R)* by software engineers and the management concerned
- The *TR phase* includes the installation and the provisional acceptance testing activities to establish that the software fulfils the requirements
- A *Software Transfer document (STD)* is produced which contains the description of the activities performed and the transfer of the software to the operation team.
- In the *OM phase*, the software is monitored for enough time to establish the final acceptance testing

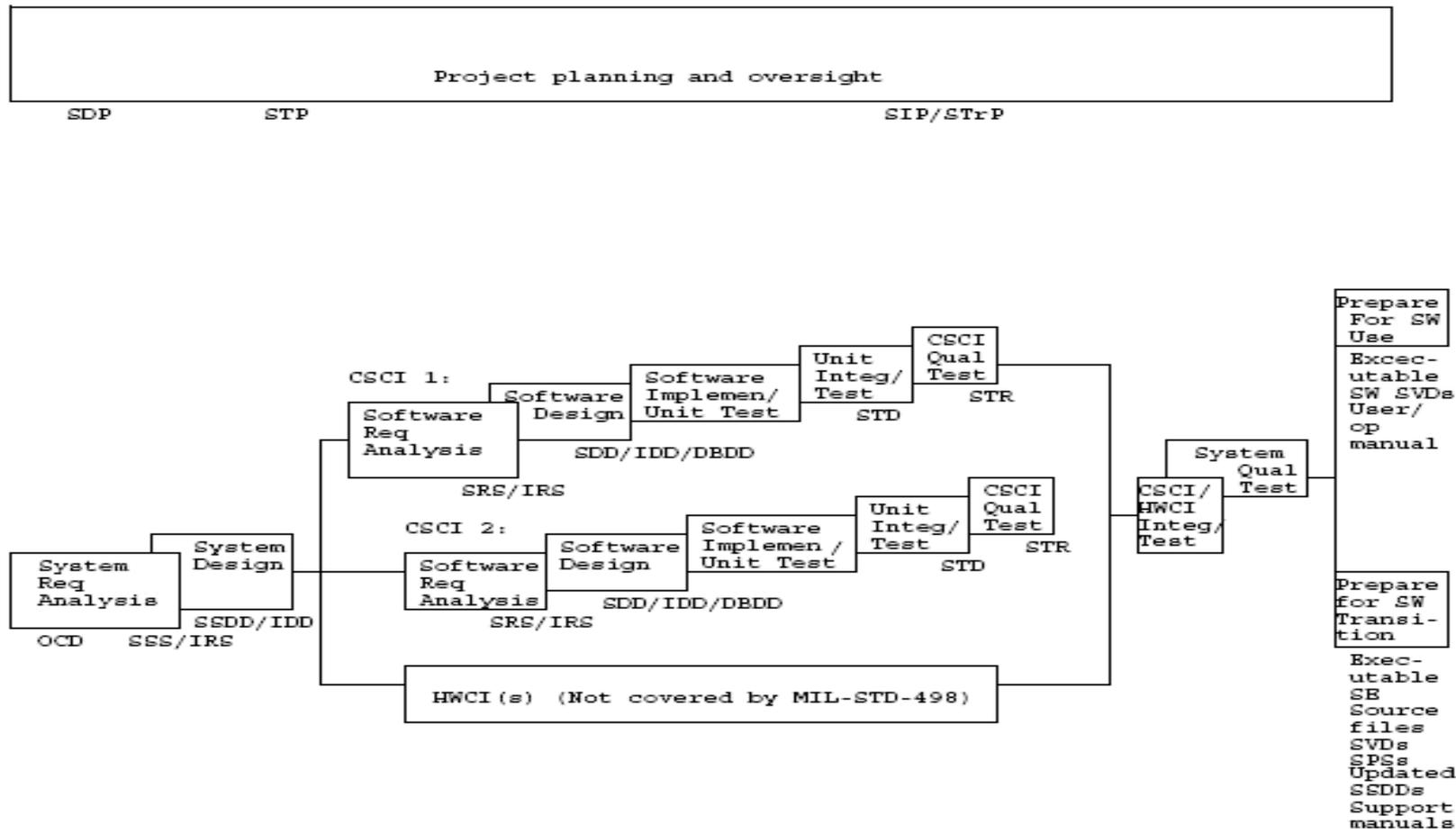
# Software Development Standards

Three standards for software development are discussed

- The software engineering standard (PSS-05-0) of the European Space Agency (ESA )
- **The MIL-STD-498 standard for software development of the US Department of Defense**
- IEEE/EIA 12207 Standard for Information Technology-software life cycle processes

# The MIL-STD-498

- The software engineering process shall include the following major activities (system view) (notes:page 2-61)
- *System requirements analysis (SYSRA) generates*
  - *Operational Concept Description document (OCD),*
  - *the System/Subsystem Specification (SSS) doc, and*
  - *the Interface Requirement Specification (IRS) doc*
- *System Design (SYSD)*, defines the system as SW/HW configuration items, and generates
  - *the System/Subsystem Design Description (SSDD) document, and*
  - *the Interface Design Description (IDD) document*



SW level environment, SW configuration management, SW product evaluation, SW quality assurance, privacy, interface with IV&V, coordination with associate developers, improvement of project processes.

Note: All activities may be more ongoing, overlapping, and iterative, than the figure is able to show.

FIGURE 2.8 Sequential Development Model (Adapted from MIL-STD 498)

# The MIL-STD-498

- Parallel development threads are then shown **for each Computer SW Configuration Item (CSCI)**
- *Software Requirements Analysis (SWRA)* generates the *SW Requirements. Specifications (SRS) and Interface Requirements Specifications (IRS) documents*
- *Software Design (SWD)* generates *Software Design Description (SDD), IDD (interfaces), and DBDD docs*
- *Software Implementation & Unit Testing (SWIUT)*
- *Unit Integration and Testing (UIT)* generates *SW Test Description (STD) doc*
- *CSCI Qualification Testing (CSCIQT)* generates *Software Test Report (STR) doc*

# The MIL-STD-498

The parallel threads of development merge into a sequential development thread in the following

- *CSCI/HWCI Integration and Testing (IT)*

generates the *system Test Description (SYSTD) doc*

- *System Qualification Testing (SYSQT)* generates the *system Test report (SYSTR)*

The last two phases are accomplished in parallel

- *Preparing for Software Use (PSWU)*
- *Preparing for Software Transition (PSWT)*

Notes: See Figures 2.8, 2.9, and 2.10 for the three models

Build 1: Establish preliminary system/software requirements and install a prototype implementing a subset of those requirements at selected user sites

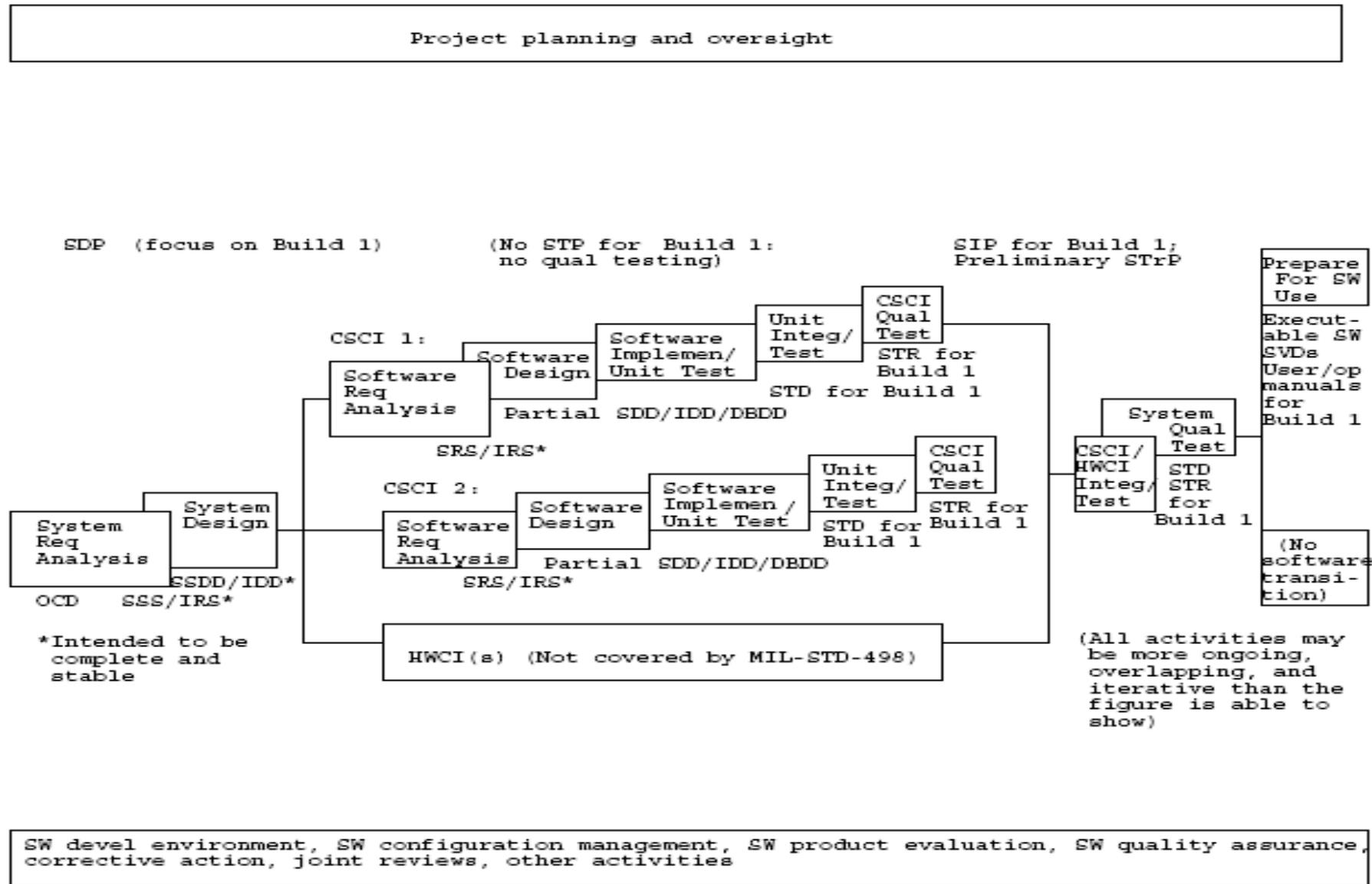
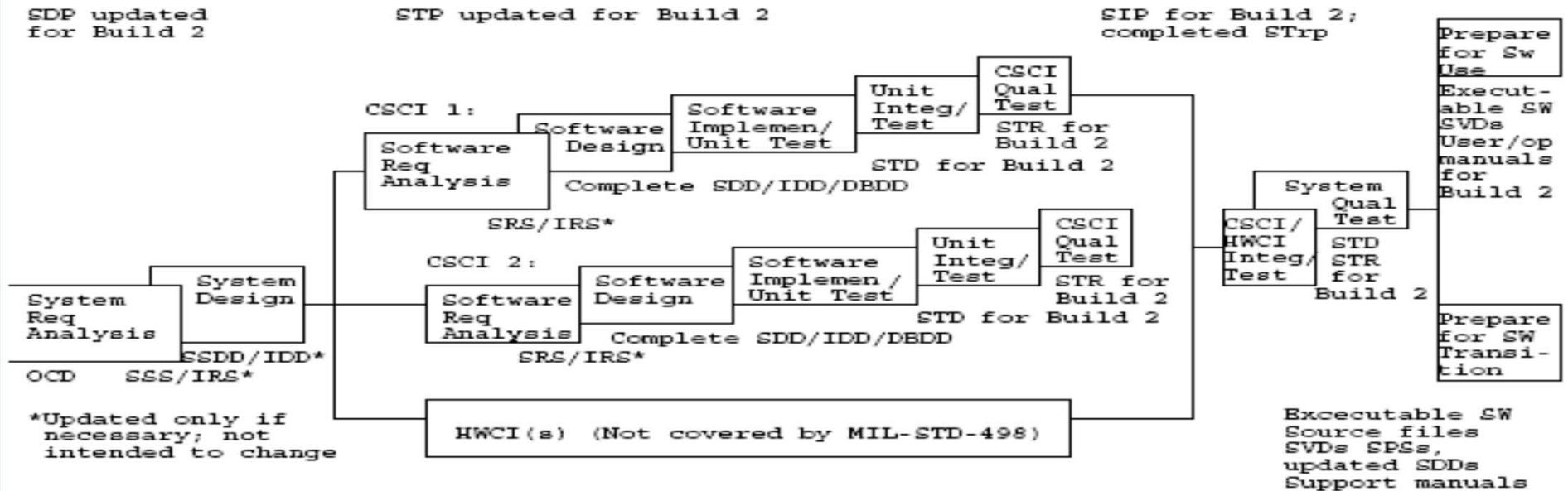


FIGURE 2.9 Incremental Model *The phases of Build 1* (Adapted from MIL-STD 498)

Build 2: Install the completed software at user sites and transition the software to the software support agency

Project planning and oversight

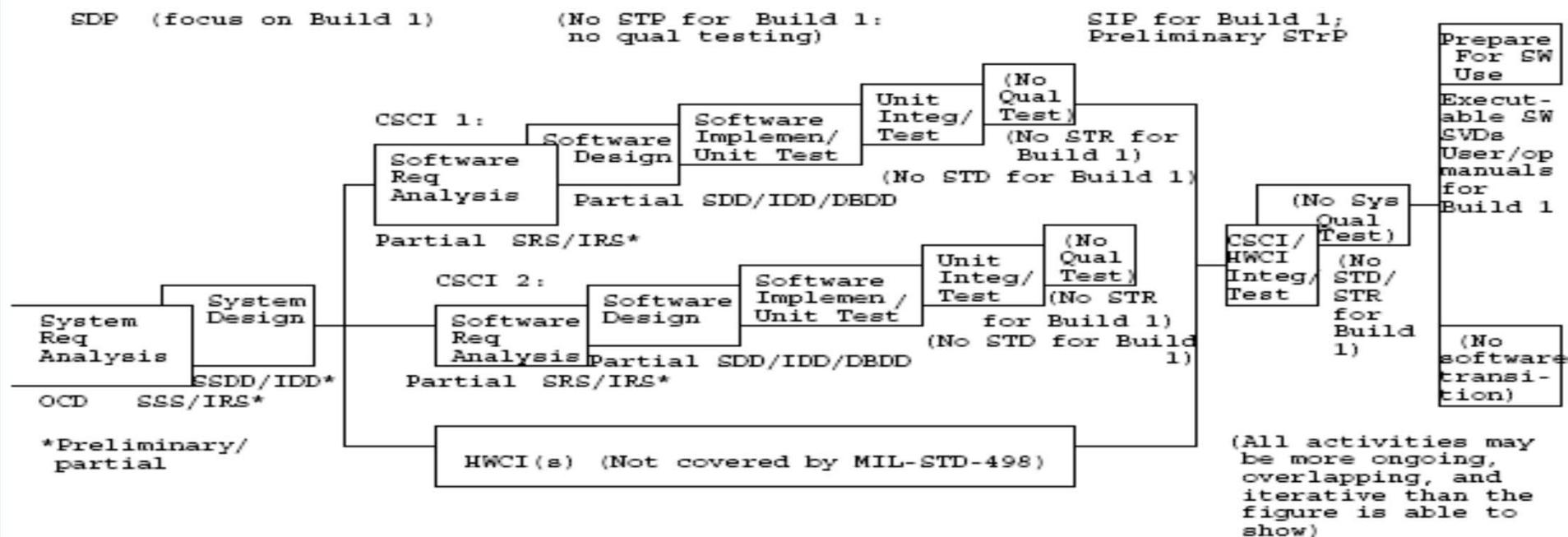


SW devel environment, SW configuration management, SW product evaluation, SW quality assurance, corrective action, joint reviews, other activities

FIGURE 2.10 Incremental Model *The phases of Build 2* (Adapted from MIL-STD 498)

Build 1: Establish preliminary system/software requirements and install a prototype implementing a subset of those requirements at selected user sites

Project planning and oversight



SW devel environment, SW configuration management, SW product evaluation, SW quality assurance, corrective action, joint reviews, other activities

FIGURE 2.11 Evolutionary Model *The phases of Build 1* (Adapted from MIL-STD 498)

# Software Development Standards

Three standards for software development are discussed

- The software engineering standard (PSS-05-0) of the European Space Agency (ESA )
- The MIL-STD-498 standard for software development of the US Department of Defense
- **IEEE/EIA 12207 Standard for Information Technology-software life cycle processes**

# IEEE/EIA 12207 Standard for Information Technology-software life cycle processes

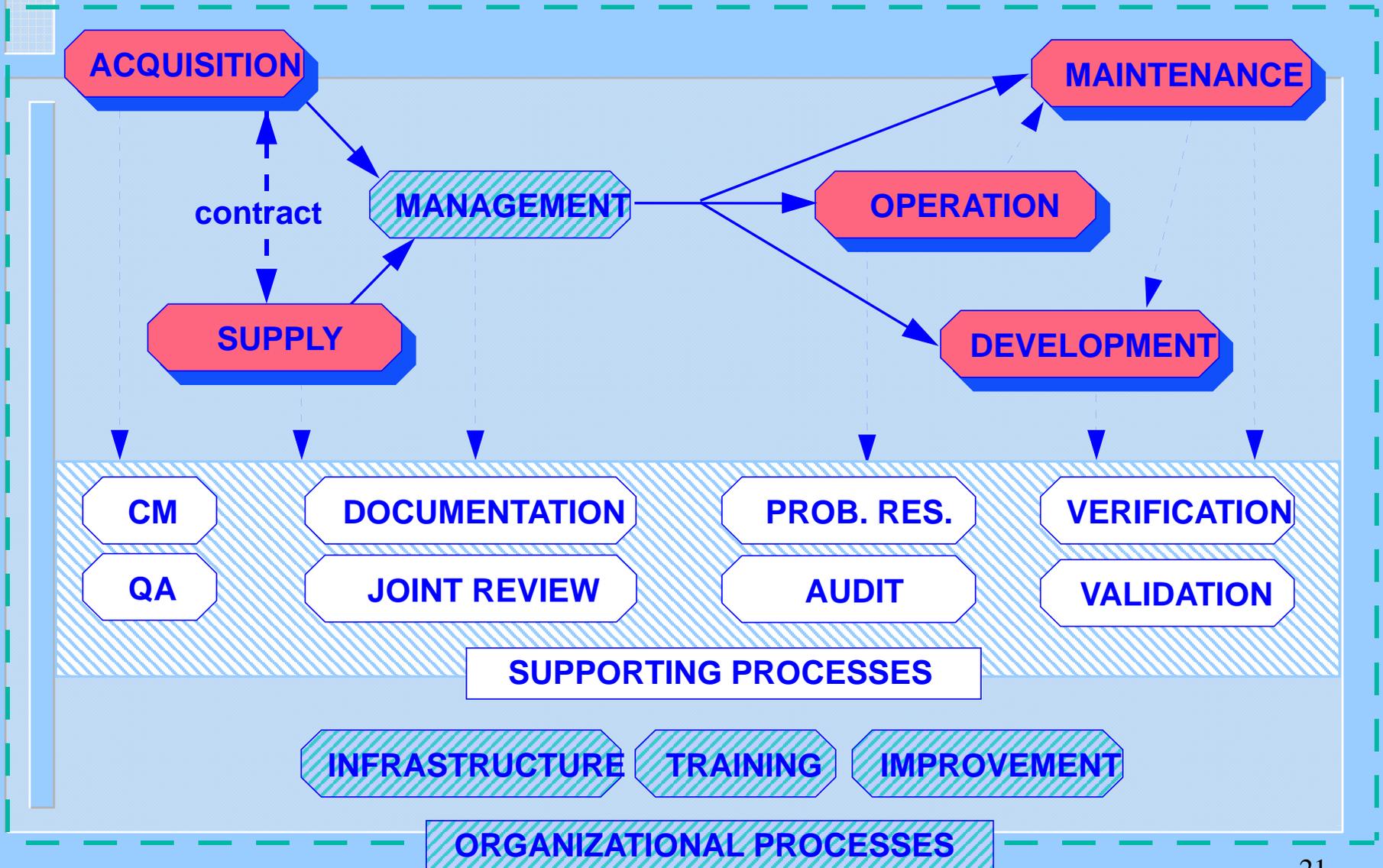
## IEEE/EIA 12207.0

### Clause 1-Scope.

**Purpose:** This International Standard establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry.

It contains processes, activities, and tasks that are to be applied during the acquisition of a system that contains software, a stand-alone software product, and software service and during the supply, development, operation, and maintenance of software products. Software includes the software portion of firmware.

# IEEE/EIA 12207 Standard for Information Technology-software life cycle processes



# The 12207 Primary Processes

## The Five Primary Processes (see Clause 5 for details)

<b>Acquisition</b>	Defines the activities of the <b>acquirer</b> , the organization that acquires a system, software product, or software service.
<b>Supply</b>	Defines the activities of the <b>supplier</b> , the organization that provides the system, software product or software service to the acquirer.
<b>Development</b>	Defines the activities of the <b>developer</b> , the organization that defines and develops the software product.
<b>Operation</b>	Defines the activities of the <b>operator</b> , the organization that provides the service of operating a computer system in its live environment for its users.
<b>Maintenance</b>	Defines the activities of the <b>maintainer</b> , the organization that provides the service of maintaining the software product; that is, managing modifications to the software product to keep it current and in operational fitness. This process includes the migration and retirement of the software product.

# The IEEE 12207 Development Process

Activity	Tasks (paraphrased)	12207.1 Information Item guidelines	
5.3.1 Process Implementation	.1 Define software life cycle model .2 Document and control outputs .3 Select and use standards, tools, languages .4 Document development plans .5 Deliver all needed products	-- -- -- Plan, Desc --	-- -- -- 6.5 DPP, 6.17 SDS --
5.3.2 System requirements analysis	.1 Specify system requirements .2 Evaluate requirements against criteria	Specification Spec, Record	6.26 SRS 6.26 SRS, 6.6 SRER
<b>5.3.3 System architectural design</b>	<b>.1 Establish top-level architecture</b> <b>.2 Evaluate architecture against criteria</b>	Description Desc, Record	6.25 SARAD 6.25 SARAD, 6.6 SAER
5.3.4 Software requirements analysis	.1 Document software requirements .2 Evaluate requirements against criteria .3 Conduct joint reviews iaw 6.6	Desc Desc, Record --	6.22 SRD, 6.30 UDD 6.22 SRD, 6.6 SRER --
<b>5.3.5 Software architectural design</b>	<b>.1 Transform requirements into architecture</b> <b>.2 Document top-level design for interfaces</b> <b>.3 Document top-level design for database</b> <b>.4 Document preliminary user documentation</b> <b>.5 Document preliminary test requirements</b> <b>.6 Evaluate architecture against criteria</b> <b>.7 Conduct joint reviews iaw 6.6</b>	Description Description Description Description Plan Desc, Record --	6.12 SAD 6.19 SIDD 6.4 DBDD 6.30 UDD 6.27 T/VP 6.12 SAD, 6.6 SAER --

<p>5.3.6 Software detailed design</p>	<p>.1 Document design for each component .2 Document design for interfaces .3 Document design for database .4 Update user documentation .5 Document unit test requirements .6 Update integration test requirements .7 Evaluate detailed design against criteria .8 Conduct joint reviews iaw 6.6</p>	<p>Description Description Description Plan Plan Rec, Desc --</p>	<p>6.16 SDD 6.19 SIDD 6.4 DBDD 6.30 UDD 6.27T/VP 6.27T/VP 6.6 DDER, 6.16 SDD --</p>
<p>5.3.7 Software coding and testing</p>	<p>.1 Document each unit, database and tests .2 Conduct and document unit testing .3 Update user documentation .4 Update integration test requirements .5 Evaluate code and test results</p>	<p>Desc, Rec, Proc Report Description Plan Rec, Plan</p>	<p>6.4 DBDD, 6.24 SCR, 6.28 T/VP 6.29 T/VRR 6.30 UDD 6.27T/VP 6.7 EOOCR, 6.6 SCTRE, 6.24SCR, 6.27T/VP</p>
<p>5.3.8 Software integratio n</p>	<p>.1 Document integration plans .2 Conduct and document integration tests .3 Update user documentation .4 Document qualification tests .5 Evaluate plans and tests against criteria .6 conduct joint reviews iaw 6.6</p>	<p>Plan, Proc Report Description Proc Proc, Desc Record, Plan</p>	<p>6.18 SIP, 6.28 T/VP 6.29 T/VRR 6.30 UDD 6.28 T/VP 6.28 T/VP, 6.30 UDD 6.6 SIER, 6.18 SIP</p>

5.3.9 Software qualification testing	.1 Conduct and document qualification testing .2 Update user documentation .3 Evaluate tests against criteria .4 Support audits iaw 6.7 .5 Prepare product for next phase	Report Description Record -- Record	6.29 T/VRR 6.30 UDD 6.6 SIER -- 6.24 SCR
5.3.10 System integration	.1 Integrate software with hardware & others .2 Document integration tests .3 Evaluate integrated system against criteria	Report Procedure Record	6.29 T/VRR 6.28 T/VPr 6.6 SQTER
5.3.11 System qualification testing	.1 Conduct and document qualification tests .2 Evaluate system against criteria .3 Support audits iaw 6.7 .4 Prepare product for installation	Report Record -- Record	6.29 T/VRR 6.6 SER -- 6.24 SCR
5.3.12 Software installation	.1 Plan installation in target environment .2 Install software iaw plan	-- --	-- --
5.3.13 Software acceptance support	.1 Support acquirer's acceptance tests .2 Deliver product per contract .3 Provide training per contract	Report Record --	6.29 T/VRR 6.24 SCR --

# The IEEE 12207 Development Process: The System Architecture Design Activity

- **5.3.3 System architectural design.** This activity consists of the following tasks, which the developer shall perform or support as required by the contract:
  - **5.3.3.1** A top-level architecture of the system shall be established. The architecture shall identify items of hardware, software, and manual-operations. It shall be ensured that all the system requirements are allocated among the items. Hardware configuration items, software configuration items, and manual operations shall be subsequently identified from these items. The system architecture and the system requirements allocated to the items shall be documented.
  - **5.3.3.2** The system architecture and the requirements for the items shall be evaluated considering the criteria listed below. The results of the evaluations shall be documented.
    - a) Traceability to the system requirements;
    - b) Consistency with the system requirements;
    - c) Appropriateness of design standards and methods used;
    - d) Feasibility of the software items fulfilling their allocated requirements;
    - e) Feasibility of operation and maintenance.

# The IEEE 12207 Development Process: The Software Architecture Design Activity

**5.3.5 Software architectural design.** For each software item (or software configuration item, if identified), this activity consists of the following tasks:

**5.3.5.1** The developer shall transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components. It shall be ensured that all the

requirements for the software item are allocated to its software components and further refined to facilitate detailed design. The architecture of the software item shall be documented.

**5.3.5.2** The developer shall develop and document a top-level design for the interfaces external to the software item and between the software components of the software item.

**5.3.5.3** The developer shall develop and document a top-level design for the database.

**5.3.5.4** The developer should develop and document preliminary versions of user documentation.

**5.3.5.5** The developer shall define and document preliminary test requirements and the schedule for Software Integration.

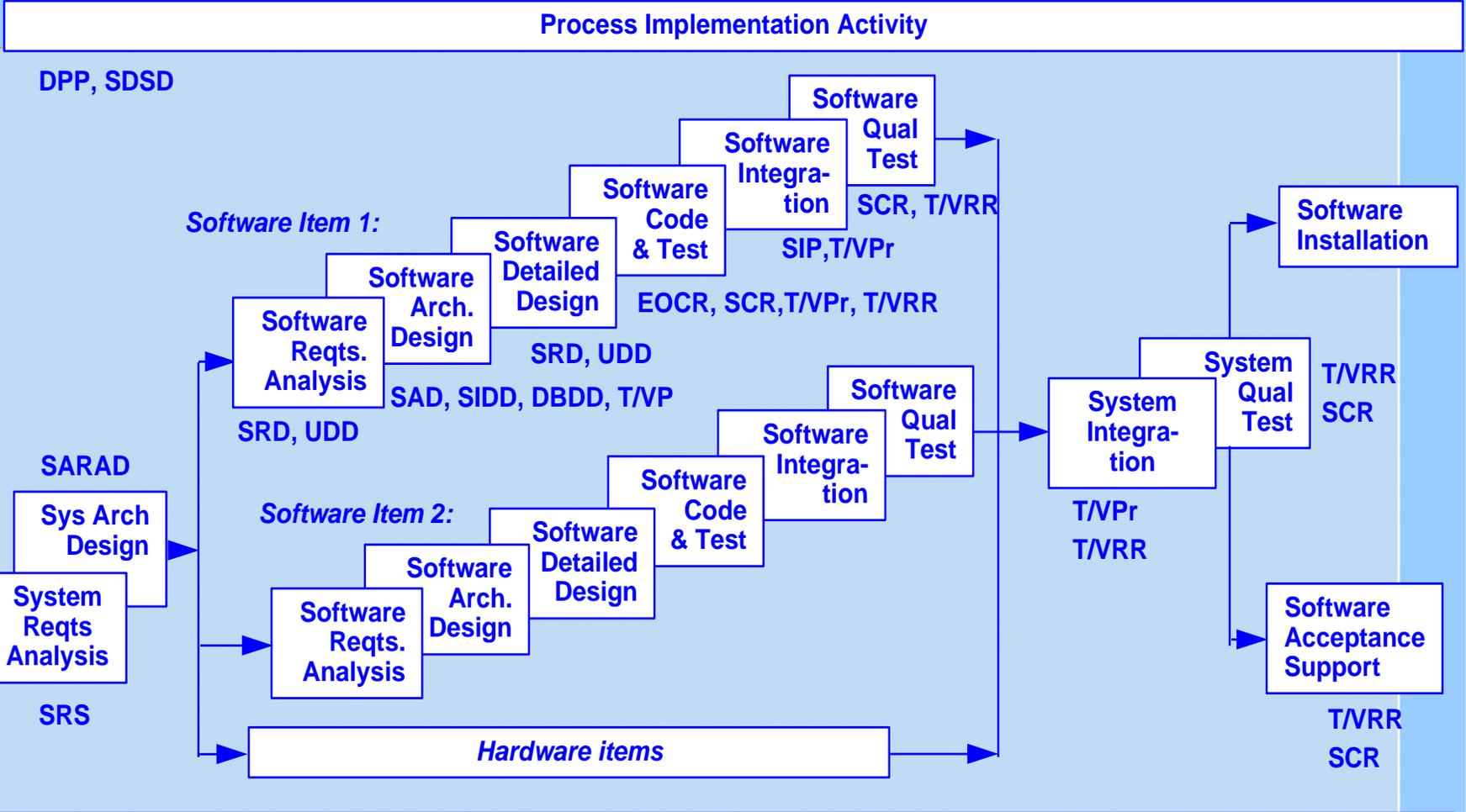
**5.3.5.6** The developer shall evaluate the architecture of the software item and the interface and database designs considering the criteria listed below. The results of the evaluations shall be documented.

- a) Traceability to the requirements of the software item;
- b) External consistency with the requirements of the software item;
- c) Internal consistency between the software components;
- d) Appropriateness of design methods and standards used;
- e) Feasibility of detailed design;
- f) Feasibility of operation and maintenance.

**5.3.5.7** The developer shall conduct joint review(s) in accordance with 6.6.

# The IEEE 12207 Development Process

One example of applying 12207 to the Waterfall development strategy



Supporting Processes: Documentation, CM, QA, Verification, Validation, Joint Review, Audit, Problem resolution

SCMP, SCMR, SCIR, SQAP, SQAR, SVRR, PR/PRR

Organizational Processes: Management, Infrastructure, Improvement, Training



# OUTLINE

- The System Life Cycle Model and the System Development Process.
- Software Engineering and the Software Development Process
- Software Development standards
- **The ICASE Environments**
- ICASE Tool: Teamwork (see notes of Ch. 2), and Software through Pictures (StP)



# The ICASE Environments

- ICASE stands for Integrated Computer-Aided Software Engineering Environments
- These environments support a variety of development techniques and notations in an integrated environment
- All the tools used in the development process are presented through a common user interface.
- Data and diagrams developed in a tool during a particular phase in the development process can be used by another tool in a different phase of development (analysis tools, design tools, etc.)

# The ICASE Environments

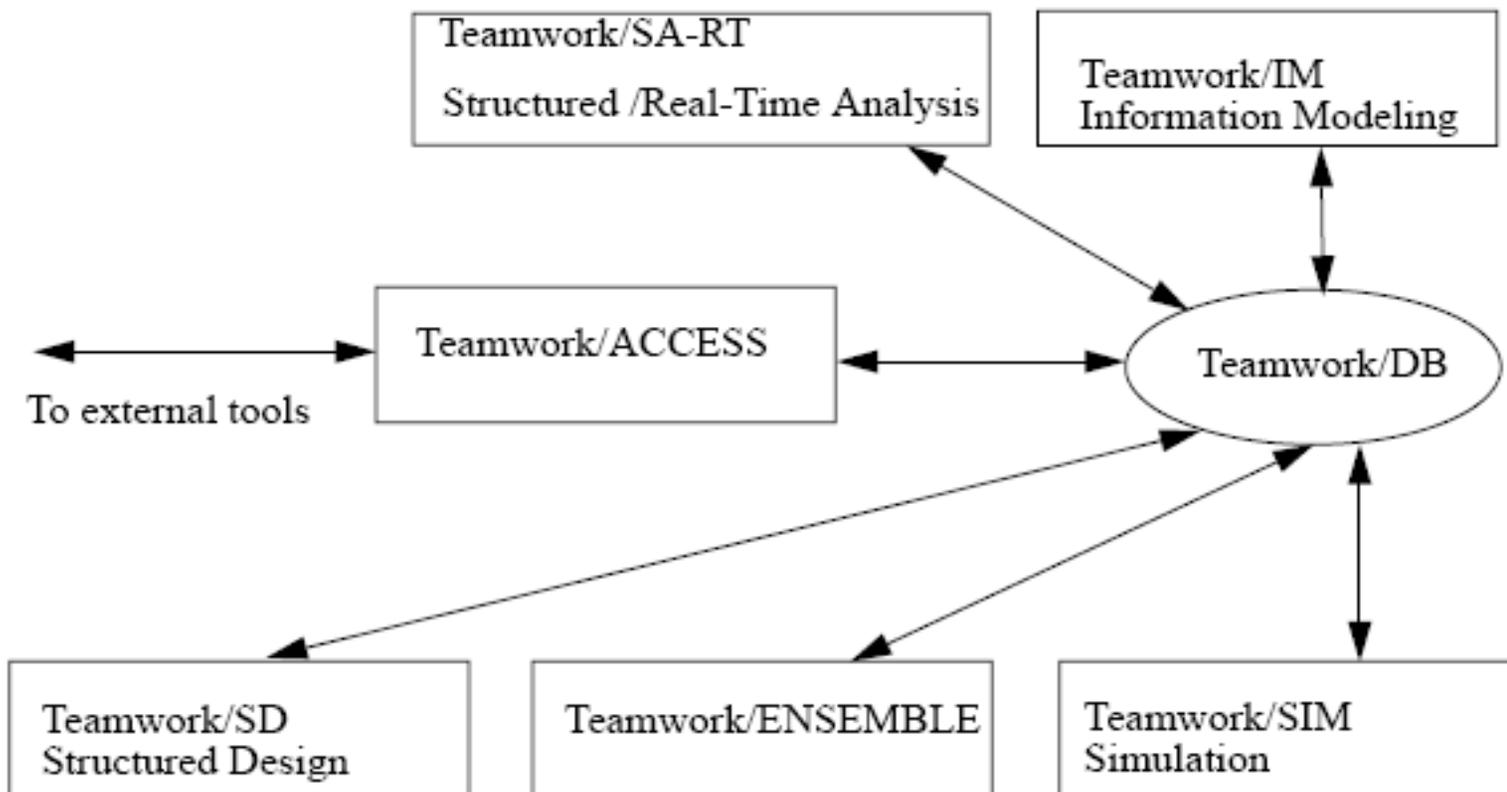


FIGURE 2.14 Example of Tightly-Coupled Data Integration Architecture