Module 14 - Software Development: Requirements and Design

Role of the Systems Engineering Plan

Effective Systems Engineering is critical to the successful development of defense systems. Within the DoD, a variety of initiatives at both the service and Office Secretary of Defense (OSD) levels have been implemented to increase overall Systems Engineering capabilities. One of these initiatives is the mandated use of a Systems Engineering Plan (SEP) by all programs.

An SEP is prepared by the Program Management Office. While the supplier typically will produce a much more technically-detailed plan, sometimes called a Systems Engineering Management Plan (SEMP), the SEP is a distinct government product. The SEP outlines the ways the acquirer will manage and control the various engineering processes used by the supplier (s), including software development processes.

Latest details on the SEP, preparation guides and SEP evaluation criteria used by the OSD are available in the lesson references.

D

Lesson Objectives

This lesson presents an overview of the Systems Engineering Process and describes how this process is used to manage the Information Technology (IT) and software development activities.

After completing this lesson, you will be able to:

- Identify Systems Engineering Technical Processes.
- Identify Systems Engineering Technical Management Processes.
- Summarize the purpose of Systems Engineering Technical Processes and Systems Engineering Technical Management Processes.

Role of Standard Processes

TOC

Standard processes offer a consistent and repeatable way to communicate between the acquirer and the developer regarding various Systems Engineering activities.

A process identifies 'what' has to be done but not the details of 'how' to do it. Standardized processes are essential for:

- Controlling the overall technical effort
- Designing systems solutions
- 'Realizing' products that make up the system

'Realization' is a formal term used to precisely quantify the desired end-state of Systems Engineering activities.

Click on the image for a formal definition of 'realization' from a Systems Engineering perspective.

D

Knowledge Review

The term 'realization' as used in Systems Engineering encompasses? (Select all that apply)

- [ ] Transitioning the product, ultimately to the customer
- [ ] Providing the physical design solution in an appropriate form
- [ ] Verifying and validating the product
- [ ] None of these responses are correct

Check Answer

Systems Engineering Processes

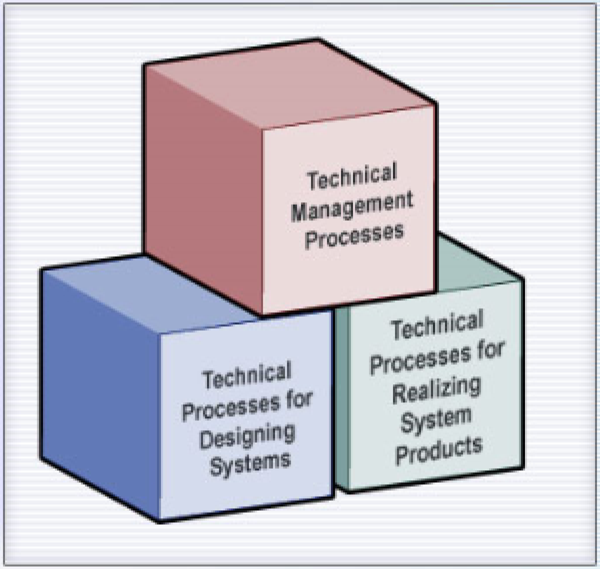
TOC

The Defense Acquisition Guidebook (DAG) supplements the DoD's 5000-series of acquisition policies. The DAG is a collection of best practices and procedures applicable across all of the phases of the Defense Acquisition Framework.

As outlined in the DAG, the DoD's Systems Engineering processes are comprised of categories consisting of:

- Technical Processes for designing systems
- Technical Processes for product realization
- Technical Management Processes

Select the above entries to see a process listing for each of these categories.



Purposes of Technical Processes

Technical Processes are used to ***design*** the products of a system including the operational products and the supporting or enabling products required to produce, support, operate, or dispose of a system as well as to ***realize*** these system products.

Select the tabs to learn more.

Designing Systems | **Realizing System Products**

Technical Processes for Designing Systems

Requirements Development — To take all inputs from users and stakeholders and translate these inputs into Technical Requirements.

Logical Analysis — To improve understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, time-related).

Design Solution — To translate the outputs of the Requirements Development and Logical Analysis process into alternative design solutions and to select a final design solution (Solution-Specified Requirements).

Knowledge Review

Which of the following is correct concerning the various Technical Processes used in Systems Engineering?

- [ ] They are used for designing systems and for product realization
- [ ] They are used to manage the overall technical effort
- [ ] 'Decision Analysis' is a key Technical Process
- [ ] Different ones are used for different types of DoD systems

Check Answer

Knowledge Review

Which of the following ***is not*** a Systems Engineering Technical Process used for system design and product realization?

- [ ] Design Solution Process
- [ ] Requirements Management Process
- [ ] Requirements Development Process
- [ ] Verification Process
- [ ] Transition Process

Check Answer

Knowledge Review

Which of the following are Systems Engineering Technical Processes used for designing systems? (Select all that apply)

- [ ] Transition Process
- [ ] Requirements Development Process
- [ ] Logical Analysis Process
- [ ] Verification Process
- [ ] Design Solution Process
- [ ] Integration Process

Check Answer

Knowledge Review

Which of the following are some of the Systems Engineering Technical Processes used for realizing system products? (Select all that apply)

- [ ] Integration Process
- [ ] Verification Process
- [ ] Validation Process
- [ ] Design Solution Process
- [ ] Requirements Development Process
- [ ] Transition Process

Check Answer

Purposes of Technical Management

Technical Management Processes are used to ***manage*** the development of system products, including supporting or enabling products.

Technical Management Processes are used in tandem with Technical Processes. The latter do the work of Systems Engineering while the former ensures the work gets done right.

Select the tabs for a summary description of each process.

Technical Management Processes | **Technical Management Processes, Cont.**

Technical Management Processes

Technical Planning — To ensure the proper application of Systems Engineering processes.

Requirements Management — To provide traceability ultimately back to user-defined capabilities and needs.

Interface Management — To ensure interface definition and compliance among the elements that compose the system; as well as with other systems with which the system or system elements must interoperate.

Risk Management — To examine the technical risks of deviating from the program plan

Knowledge Review

Which of the following is correct concerning the use of the various Technical Management Processes used in Systems Engineering?

- [ ] They are used extensively in system design
- [ ] Product "realization" uses them to field a product
- [ ] "Design Solution" is the most important of the Technical Management Processes
- [ ] They are key to managing the overall technical effort
- [ ] None of these responses is correct

Check Answer

Knowledge Review

Each of the following is a Systems Engineering Technical Process used for designing systems **except** for:

- [ ] Requirements Development Process
- [ ] Logical Analysis Process
- [ ] Technical Planning Process
- [ ] Design Solution Process

Check Answer

Knowledge Review

Which of the following is correct concerning the various Technical Processes used in Systems Engineering?

- [ ] They are used for designing systems and for product realization
- [ ] They are used to manage the overall technical effort
- [ ] 'Decision Analysis' is a key Technical Process
- [ ] Different ones are used for different types of DoD systems

Check Answer

Knowledge Review

Each of the following is a Systems Engineering Technical Management Process except: (Select all that apply)

- [ ] Requirements Development Process
- [ ] Transition Process
- [ ] Decision Analysis Process
- [ ] Interface Management Process
- [ ] Technical Planning Process

Check Answer

Knowledge Review

Which of the following best defines Requirements Management, one of the Systems Engineering Technical Management processes?

- [ ] To provide traceability ultimately back to user-defined capabilities.
- [ ] To plan for, acquire, access, manage, protect, and use data of a technical nature.
- [ ] To examine the technical risks of deviating from the program plan.
- [ ] To provide the basis for evaluating and selecting alternatives when decisions need to be made.

Check Answer

The Big Picture I: Product Design Snapshot

Let's look at a simplistic (you'll find out why later!) example of Systems Engineering Technical Processes in action including some of the terminology used. Starting with design processes which are outlined here as:

1. **Requirements Development Process:** Requirements for DoD systems come from various diverse constituencies (generically categorized process-wise, as Acquirer Requirements and Other Interested Party Requirements). These requirements comprise the set of Stakeholder Requirements which include Derived Requirements. These are ultimately transformed into Technical Requirements. The latter, when baselined, become the basis for the design. Key Measures of Effectiveness are also initially identified.
2. **Logical Analysis Process:** Technical Requirements are then analyzed in various ways to determine an optimal functional architecture. Interfaces are defined. Performance parameters are allocated. Important additional Technical Requirements (called Derived Technical Requirements) are identified. Work products are baselined.
3. **Design Solution Process:** Using the outcomes from the Requirements Development Process and the Logical Analysis Process, alternative design solutions are developed. More Derived Technical Requirements may be identified as well. These design solutions are evaluated to determine which are acceptable within cost, schedule and technical constraints. A design or a physical architecture is established.

Outcomes of the Design Solution Process typically include a set of Solution-Specified Requirements, a key component of which is the System Specification. These requirements are baselined and become part of a Technical Data Package (TDP).

The Big Picture II: Product Realization Snapshot

Realization now begins with these Technical Processes:

4. **Implementation Process:** If subsystems do not need to be developed, then using appropriate elements from the TDP, the product is made, bought or reused. This may involve hardware fabrication, manufacturing, software coding or purchase from outside sources. No matter what their source, some degree of low-level verification and validation is performed to make sure that 'good' products get used for implementation. Some supporting documentation may be developed.
5. **Integration Process:** At some point implementation is completed. End product components are assembled and integrated. Prior to this, implemented components are verified and validated as appropriate so as to ensure that only 'good' products actually get assembled and integrated.
6. **Verification Process:** Using criteria established as part of the Design Solution Process and various plans, the end product is verified that it conforms to its 'design-to' requirements. Has it been 'built right'?
7. **Validation Process:** Using criteria established as part of the Design Solution Process and various plans, the end product is validated that it conforms to its Stakeholder Requirements. Is it the 'right product'?
8. **Transition Process:** The end product either: (1) moves up a level (e.g., from subsystem to system) in the physical architecture of the system for more development as needed; (2) it transitions to the next Defense Acquisition Framework phase, or (3) the End Product is delivered successfully to the user.

Simple, eh? Wait! Not so fast...

Not So Fast

TOC

The Technical Process snapshots, just provided, while correct from a big picture standpoint, are simplistic from several actual application perspectives:

- **Non-Linear:** The processes were depicted as occurring in a strict linear sequence. This is not typical: there is a high degree of healthy interaction between them, especially among the design processes. This interaction is in the form of iteration (do it until you get it right!) and recursion (keep doing it until you are done!).
- **Entire DAF:** The processes were depicted as just from a later stage in the Defense Acquisition Framework (DAF). In actual practice, these Technical Processes will occur repeatedly during each acquisition phase, but with different types of realized products.
- **Complex System Model:** The processes were depicted as operating on a simple model structure. In practice, for non-trivial systems, the system structure consists of many layers and levels. Throughout this layered structure, multiple Technical Processes are occurring simultaneously.

So...a lot of things are going on at the same time in different parts of the system structure being done by many different organizations. Given this, it's all too easy to make chaos out of order!

That is why Technical Management Processes are absolutely vital complements to the Technical Processes.

Knowledge Review

What ordering of Technical Processes will complete this process list in the best order? ______________, Logical Analysis, ________________, Implementation

- [ ] Transition, Validation
- [ ] Verification, Validation
- [ ] Requirements Development, Design Solution
- [ ] Requirements Management, Interface Management

Check Answer

Knowledge Review

What ordering of Technical Processes will complete this process list in the best order? Integration, ______________, ______________, ____________.

- [ ] Transition, Validation, Verification
- [ ] Validation, Verification, Transition
- [ ] Logical Analysis, Requirements Development, Design Solution
- [ ] Implementation, Verification, Validation
- [ ] Verification, Validation, Transition

Check Answer

The Big Picture III: Control Processes

Continuing our example, let's look at the application of the various Technical Management Processes. Five of these are generally referred to as ***Technical Control Processes***. These are:

1. **Requirements Management Process**: Used heavily as design processes iterate and interact. It ensures that all of the generated Technical Requirements and Derived Requirements can ultimately be traced back to user-defined capabilities and needs.
2. **Interface Management Process**: Used to support the Logical Analysis Process where many interfaces are defined as well as the Integration Process. Interface Management ensures interface definition and compliance among system elements and for interoperability with other systems.
3. **Risk Management Process**: Is a core process which underlies all others. It Identifies, Analyzes, Mitigates and Tracks program technical risks.
4. **Configuration Management Process**: Ensures that baselines are established, controlled and kept consistent. Configuration Management is a key player with Requirements Management, Interface Management and Technical Data Management Processes.
5. **Technical Data Management Process**: A wide variety of data and information is produced as outcomes of various process activities. Technical Data Management plans for it; acquires it; provides access to it; manages it; and protects data and information of a technical nature needed to support the total life cycle of a system.

The Big Picture IV: Other Technical Management Processes

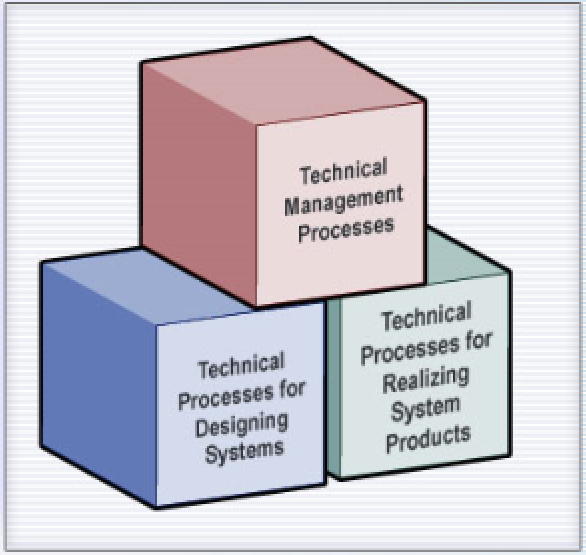
TOC

Continuing:

6. **Technical Planning Process** : This management process is the lynchpin of the entire Systems Engineering effort. Various key plans, such as the Systems Engineering Plan (SEP), which outlines expected activities by acquisition phase; the Test and Evaluation Master Plan (TEMP), and various other plans, are key outcomes of the Technical Planning Process.
7. **Technical Assessment Process** : Gives visibility to 'where we are' and 'where we are going' regarding the application of Technical Processes. Key tools used in Technical Assessment include Technical Reviews, Earned Value Management (EVM) and Technical Performance Measurement (TPM) .
8. **Decision Analysis Process** : A cross-cutting process that provides a rational, repeatable basis for evaluating and selecting alternatives when a decision must be made. Operates within the program's Trade Space.

Activities, Tasks, and Outcomes

The 16 technical and technical management processes outlined previously are the foundations of the DoD's approach to Systems Engineering and technical management. Additional details can be found in the Defense Acquisition Guidebook (DAG).



Other topics within other lessons break down each of these eight Technical Processes and eight Technical Management Processes. They are broken down into over 90 process-specific activities, each with associated expected outcomes. Each activity is further decomposed into more detailed sets of illustrative tasks.

These illustrative tasks put the process-specific activity into perspective. They show one possible way the activity could be accomplished and its expected outcomes used.

D

Process Activity Roadmap

TOC

With all these activities, tasks and expected outcomes, one can get overwhelmed without some sort of a guide. Looking at them all from a macro level, many of the activities and associated tasks making up a process follow a generic roadmap. The activity roadmap generally will look like some variation of the following:

- **GET READY**: Prepare to do the activity
- **DO IT**: Perform process activities
- **ADJUST AND ANALYZE**: As needed
- **BASELINE**: Baseline results in some manner
- **CAPTURE**: Update the program's technical management database

While this simplifies things somewhat, nevertheless, there is a lot going on ***and a lot that needs to go on***, in Systems Engineering! This is why numerous studies and 'lessons learned' have shown that the levels of investment in program technical management are very good predictors of later overall program success.

Review of Objectives

TOC

Systems Engineering Technical Processes for designing systems include:

1. **Requirements Development**: Translating stakeholder needs into Technical Requirements
2. **Logical Analysis**: To improve understanding of requirements and their relationships
3. **Design Solution**: To develop alternative design solutions and select a final design (Solution-Specified Requirements)

Systems Engineering Technical Processes for realizing systems products include:

1. **Implementation**: Creating (making, buying or reusing) low-level system elements
2. **Integration**: Incorporation of lower-level system elements into higher level ones
3. **Verification**: Confirming system elements meet design-to or build-to specifications
4. **Validation**: Confirming system elements meets Stakeholder Requirements
5. **Transition**: Moving a system element to the next development stage or, for the End Product, to the user

Review of Objectives, Cont.

TOC

Systems Engineering ***Technical Management Processes*** include:

1. **Technical Planning:** Ensuring the proper application of Systems Engineering processes
2. **Requirements Management:** Providing traceability of system and subsystem requirements ultimately back to user-defined capabilities and needs
3. **Interface Management:** Ensuring interface definition and compliance among system elements including other systems
4. **Risk Management:** Examination of the technical risk of deviating from program plans
5. **Configuration Management:** The establishment and maintenance of the consistency of a product's attributes with its requirements and configuration information
6. **Technical Data Management:** Planning for, acquiring, accessing, managing, protecting and using data of a technical nature for supporting the total life cycle of a system
7. **Technical Assessment:** Measuring technical progress and the effectiveness of plans and requirements
8. **Decision Analysis:** Provides the basis for evaluation and selection of technical alternatives when decisions need to be made

Summary

This lesson presented an overview of the Systems Engineering Process and how it is used to manage the Information Technology (IT) and software development activities.

Systems Engineering uses a multidisciplinary approach to tie together all aspects of a project, making sure that individual parts, subsystems, support equipment and associated operational equipment effectively function as intended in the operational environment.

Lesson Completion

You have completed the content for this lesson.

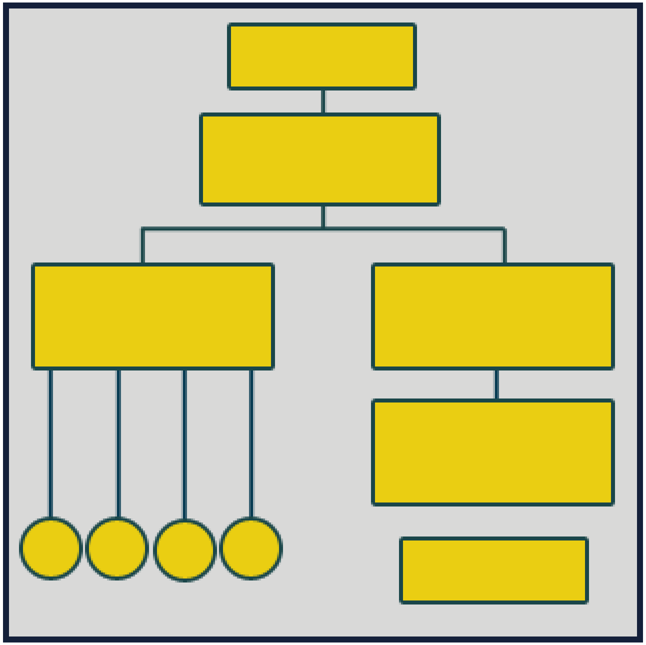
To continue, select another lesson from the Table of Contents on the left.

If you have closed or hidden the Table of Contents, click the Show TOC button at the top in the Atlas navigation bar.

Work Breakdown Structure

TOC

A Work Breakdown Structure (WBS) is used in systems development to document the product being developed and its associated processes, data and facilities.



Lesson Objective

This lesson presents a more detailed look at the system decomposition process. The emphasis is on the software component of a typical WBS.

After completing this lesson, you will be able to:

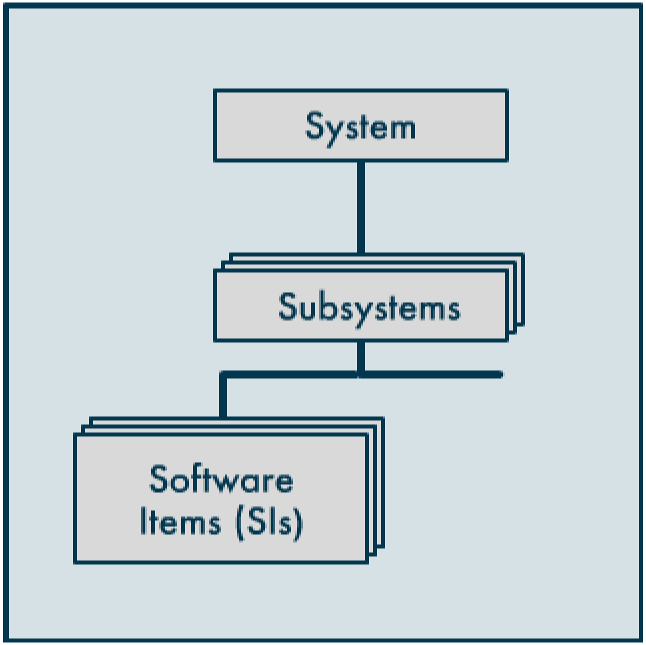
- Recognize the software component of a WBS.

WBS: Three Levels

At the **system** level, top-level aggregate system requirements and performance goals are established.

As part of the Systems Engineering Process, highly complex systems are broken down into smaller **subsystems**. This continues until ultimately the system/subsystems are decomposed into discrete products.

As part of this process, software requirements are identified and assigned to particular pieces of software called **Software Items**.



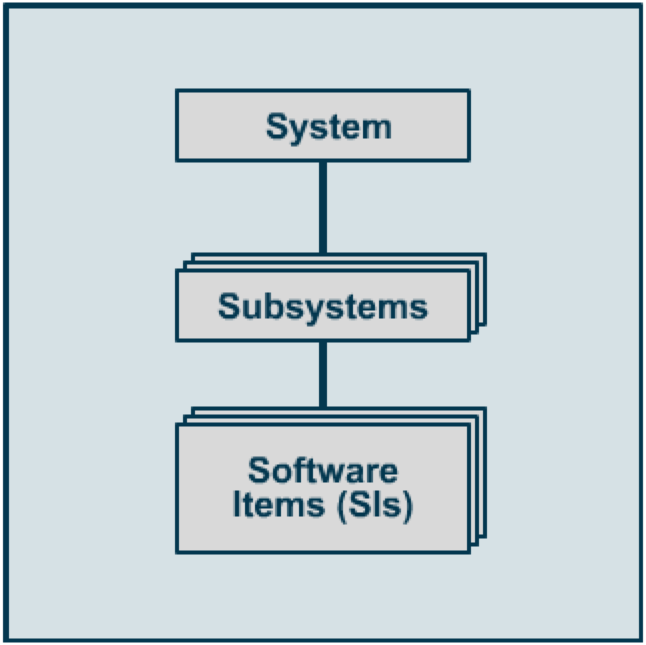
D

WBS: Software Items (SIs)

TOC

A Software Item (SI) is a grouping of software that satisfies an end-use function. It is specifically designated as an SI for purposes of specification, qualification testing, interfacing or other purposes.

When identified for configuration management purposes, the term Software Configuration Item (SCI) is used.

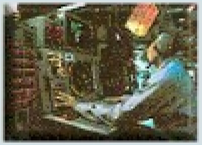


D

SI Breakout Examples

TOC

Software for an Active/Passive Sonar System



Software for a Large-Scale Submarine Combat Control System

Software for an Automated Message Handling System

To see examples of Software Item breakouts used on actual systems, please select each photo shown here

Knowledge Review

Software requirements are identified and assigned to particular pieces of software. These pieces are called ______.

- [ ] Software Units
- [ ] Subsystems
- [ ] Aggregate systems
- [ ] Software Items

Check Answer

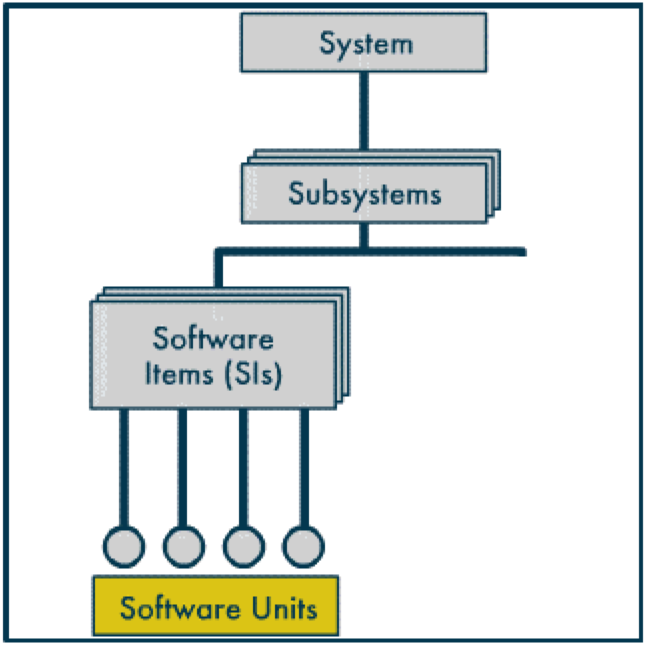
IPT Member Perspectives: Software Items

Before continuing select each LRATS IPT team member to learn about their perspective on how the choice of Software Items can impact their functional area.

Software Units

Software Items are still too large to design and code directly. Some Software Items can be as large as 100,000 source lines of code (SLOC).

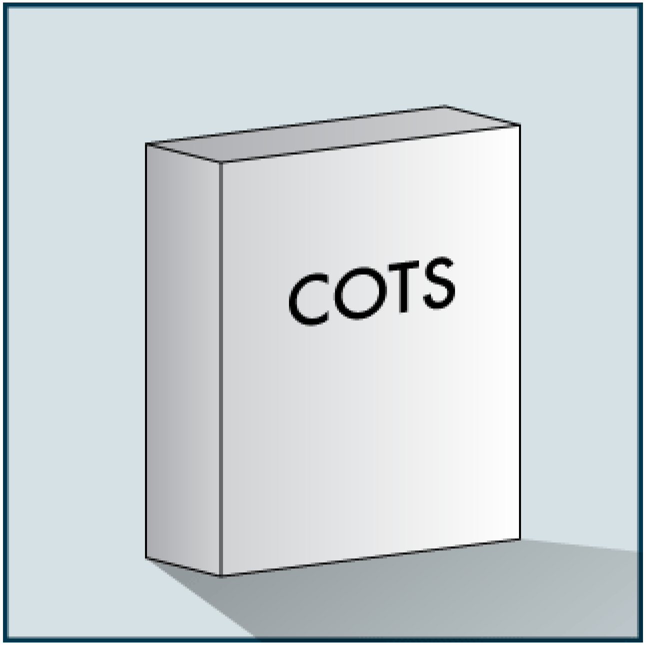
So, requirements allocation to smaller "pieces" (modules, components, etc.) of the SI continues as part of software design. Ultimately, an SI is designed and broken down into the lowest level component. This is called a Software Unit. Each Software Unit is then designed and coded.



Using COTS

As an alternative to full-scale design and coding, Commercial-Off-the-Shelf (COTS) products may be used to satisfy a set of requirements at the Software Item or Software Unit levels.

For those systems that can employ COTS, some development steps may not be necessary. Other steps still play critical roles in effective COTS employment.

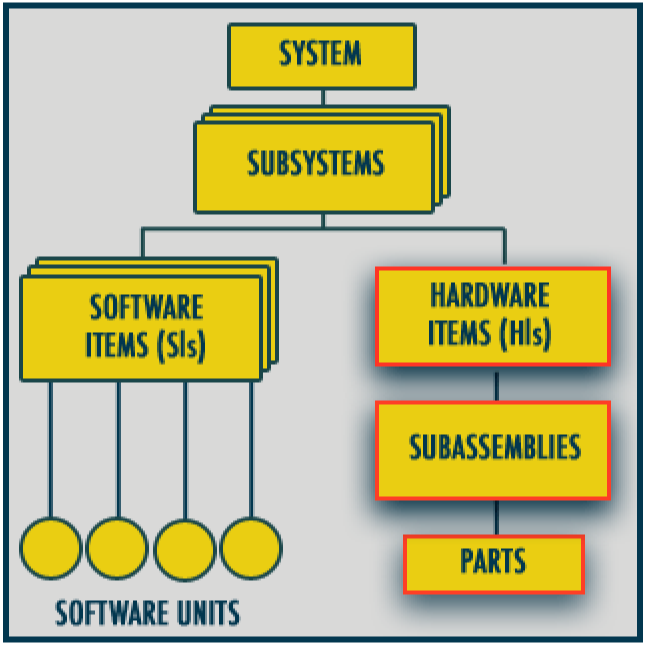


Hardware Items (HIs)

TOC

Similar to Software Items, the hardware for a system is broken into products. These are called Hardware Items (HIs).

Just as SIs are partitioned, a similar partitioning process applies to HIs. For example, HIs can be broken down into subassemblies and then into parts.



D

Requirements Traceability

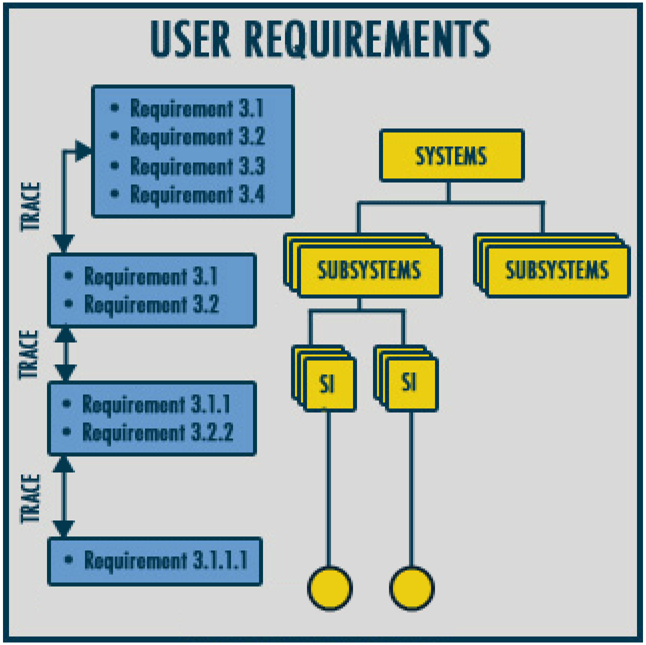
TOC

At every point in the decomposition of a system, careful attention is paid to tracing requirements.

This traceability process:

- Ensures that no system-level requirement is overlooked
- Prevents gold-plating
- Identifies derived requirements
- Helps to ensure that test coverage is complete

In short, traceability ensures that the whole equals the sum of its parts.



D

Knowledge Review

Which of the following appears at the top level of a WBS?

- [ ] System
- [ ] Software Items
- [ ] Software Units
- [ ] Hardware Items

Check Answer

Summary

This lesson covered the Work Breakdown Structure (WBS). The WBS is used in systems development to document the product being developed and its associated processes, data and facilities.

There are three levels to a WBS:

- Systems
- Subsystems
- Software Items (SIs)

Software Items (SIs) are further broken down into smaller pieces (modules, components, etc.) called a Software Unit.

Commercial-Off-the-Shelf (COTS) products may be used to satisfy a set of requirements at the Software Item or Software Unit levels.

Lesson Completion

You have completed the content for this lesson.

To continue, select another lesson from the Table of Contents on the left.

If you have closed or hidden the Table of Contents, click the Show TOC button at the top in the Atlas navigation bar.

Life Cycle Reviews and Documentation

Life cycle reviews and documentation are a critical part of the development management process for an IT software-intensive system.

This lesson provides examples of life cycle reviews and documentation that can be used as part of the development management process for an IT software-intensive system.

Once you have completed this lesson, you will be able to define life cycle reviews and documentation that can be used to help manage software development.

Lesson Objective

After completing this lesson, you will be able to:

- Define life cycle reviews and documentation used to help manage software development.

Tailoring Reviews and Documentation

Two important points about life cycle reviews and documentation:

- DoD Project Offices have wide latitude in tailoring these activities and items to the needs of a specific project.
- The actual choice and timing of the reviews and documentation used depends on the acquisition strategy, the specific software development paradigm employed and the overall software support strategy for the project.

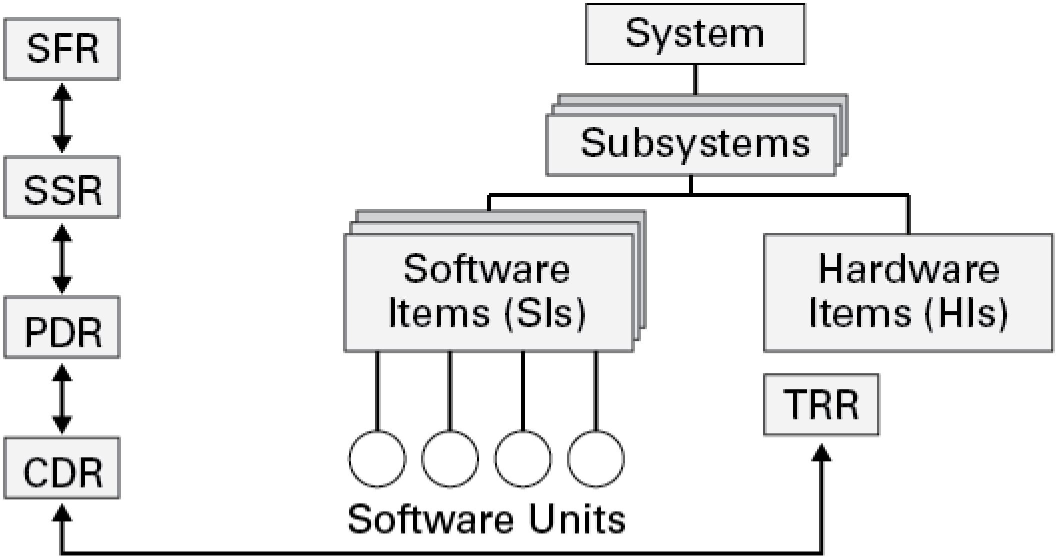
Life cycle reviews and documentation link to the Integrated AT&L Life Cycle Management Framework.

Life Cycle Reviews

Life Cycle Reviews assess the progress at the various stages of development. Integrated Product Teams (IPTs) and the collaboration process play key roles in the life cycle review process.

Some of the major reviews that can be used for software systems include: System Functional Review (SFR), Software Specification Review (SSR), Preliminary Design Review (PDR), Critical Design Review (CDR), and Test Readiness Review (TRR).

Select SFR, SSR, PDR, CDR, and TRR in the graphic to learn more.



D

Key Life Cycle Documentation

An integral part of software development is Life Cycle Documentation. Key life cycle documentation used in software programs include: System/Subsystem Specification (SSS), Software Requirements Specification (SRS), Interface Requirements Specification (IRS), Software Design Description (SDD), and Interface Design Description (IDD).

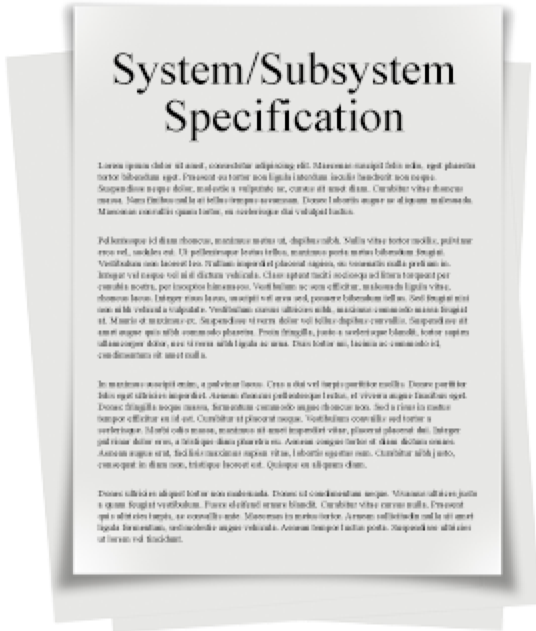
Select the tabs to learn more about key life cycle documentation used in software programs.

SSS	SRS	IRS	SDD	IDD

The System/Subsystem Specification (SSS) represents the translation of operational needs into system-level technical requirements.

The SSS:

- Documents the functional baseline for the system or subsystem
- Is the basis for top-level design
- Includes qualification methods
- Specifies top-level requirements



Knowledge Review

Which of the following life cycle documentation identifies interface requirements for a Software Item?

- [ ] System/Subsystem Specification
- [ ] Interface Requirements Specification
- [ ] Software Requirements Specification
- [ ] Interface Design Description

Check Answer

Knowledge Review

Which life cycle documentation documents design decisions for a Software Item (SI)?

- [ ] System/Subsystem Specification
- [ ] Software Requirements Specification
- [ ] Interface Requirements Specification
- [ ] Software Design Description

Check Answer

Knowledge Review

Which life cycle documentation specifies top-level requirements?

- [ ] System/Subsystem Specification
- [ ] Interface Requirements Specification
- [ ] Interface Design Description
- [ ] Software Requirements Specifications

Check Answer

Summary

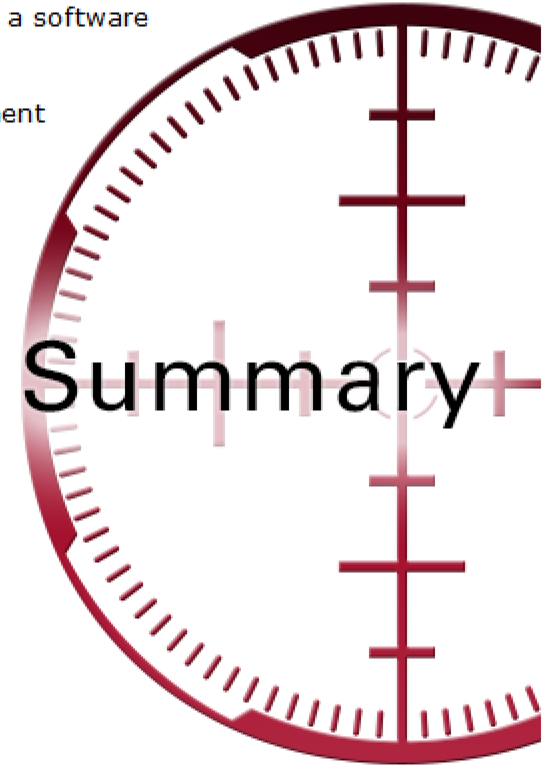
This lesson provided examples of Life Cycle Reviews and Documentation that can be used as part of the development management process for a software system.

There are five types of reviews generally used in the IT development management process:

- System Functional Review (SFR)
- Software Specification Review (SSR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Test Readiness Review (TRR)

Principle documentation includes:

- System/Subsystem Specification (SSS)
- Software Requirements Specification (SRS)
- Interface Requirements Specification (IRS)
- Software Design Description (SDD)
- Interface Design Description (IDD)



Lesson Completion

You have completed the content for this lesson.

To continue, select another lesson from the Table of Contents on the left.

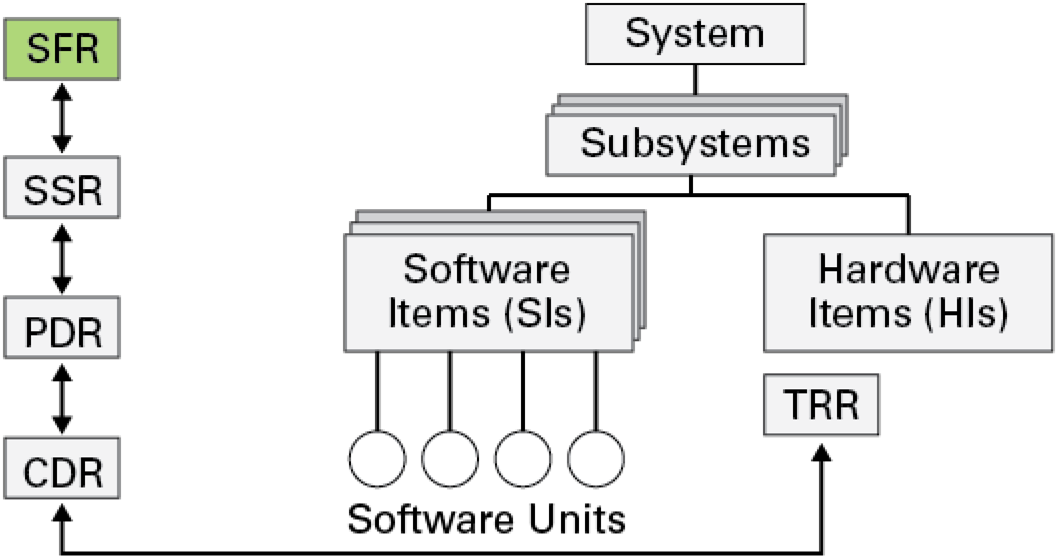
If you have closed or hidden the Table of Contents, click the Show TOC button at the top in the Atlas navigation bar.

System Functional Review (SFR)

The System Functional Review (SFR) is a system-level review. As part of the SFR, reviewers:

- Evaluate optimization, traceability, completeness and risk of the system-level design
- Review supplier progress in defining design requirements for products below the system or subsystem level

Normally the SFR is the point in the life cycle at which the acquirer approves the System/Subsystem Specification (SSS).



D

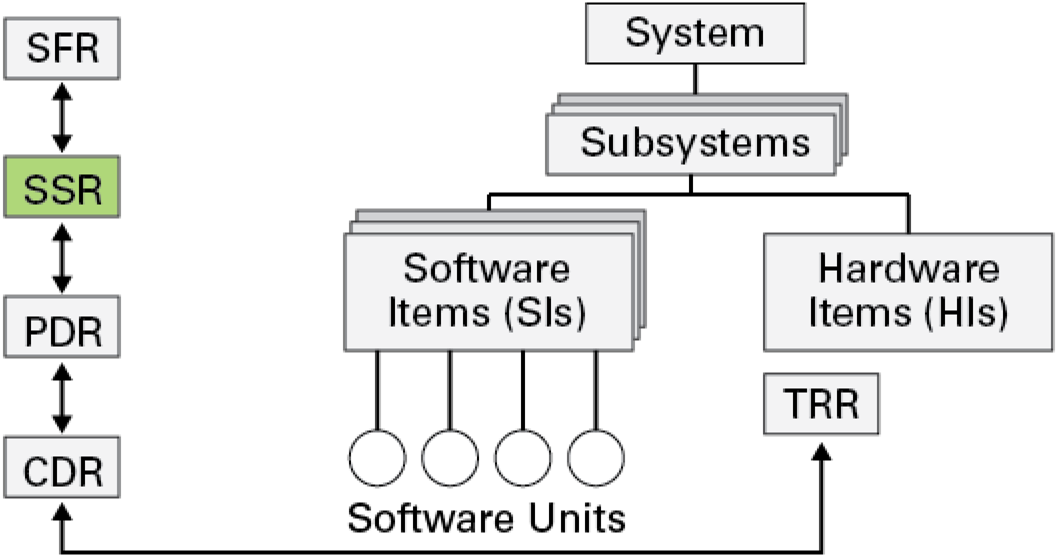
Software Specification Review (SSR)

TOC

A Software Specification Review (SSR) is typically held for each Software Item (SI). During the SSR, reviewers:

- Examine the quality and completeness of software requirements, including derived requirements, for a particular SI
- Assess the readiness to start software design activities

Typically the SSR results in approval of the Software Requirements Specification (SRS) and the Interface Requirements Specification (IRS).



D

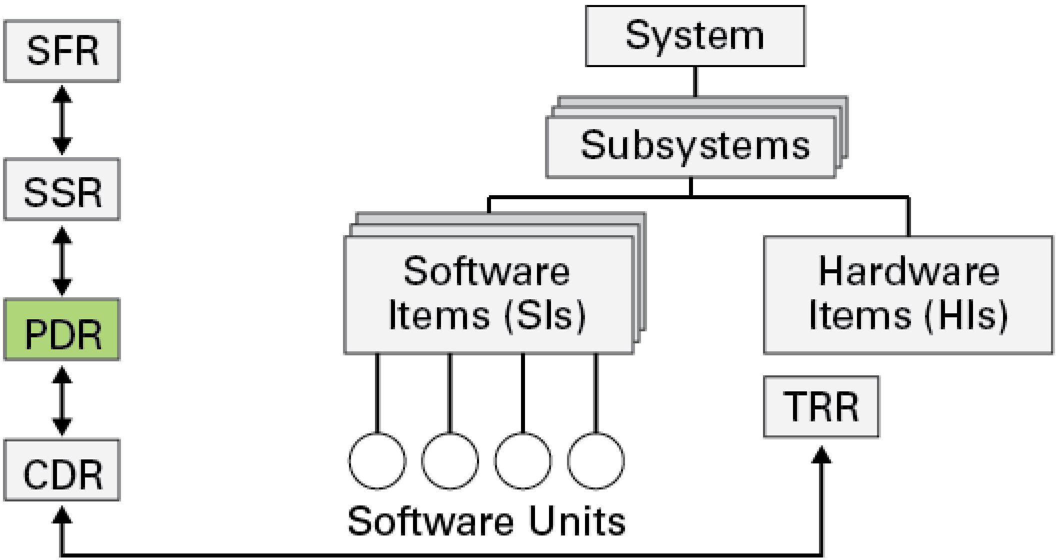
Preliminary Design Review (PDR)

TOC

The Preliminary Design Review (PDR) for software is focused at the Software Item level. PDRs are:

- Conducted after top-level design efforts are completed, but before detailed design has begun
- Held for each Software Item or a functionally related group of them

Successful completion of PDR indicates readiness to start the detailed design.



D

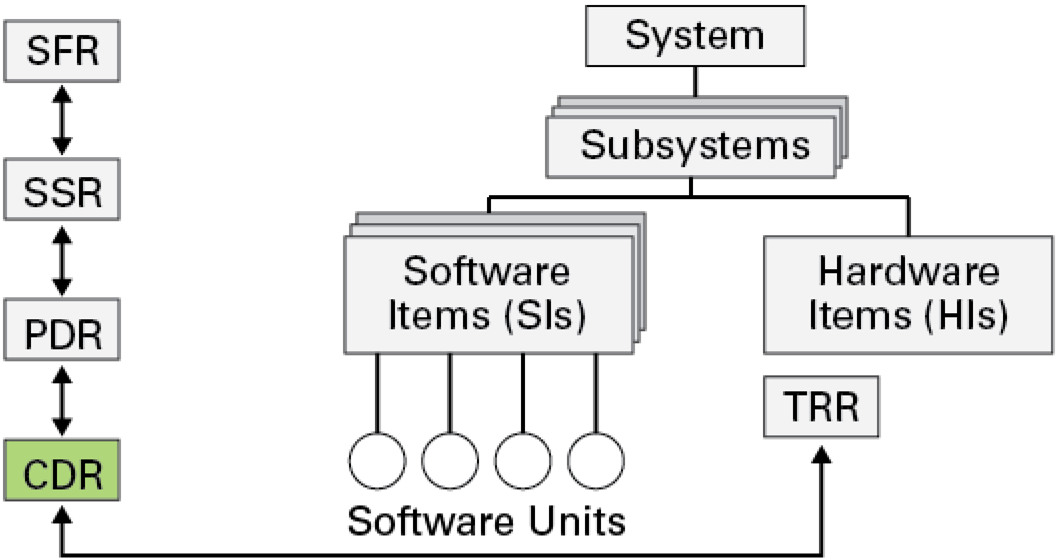
Critical Design Review (CDR)

TOC

The Critical Design Review (CDR) for software is the last major **design** review. During the CDR, reviewers examine the detailed design to establish its integrity prior to coding and testing.

Typically the CDR:

- Results in final approval of the Software Design Description (SDD) and Interface Design Description (IDD)
- Initiates development of final versions of source code

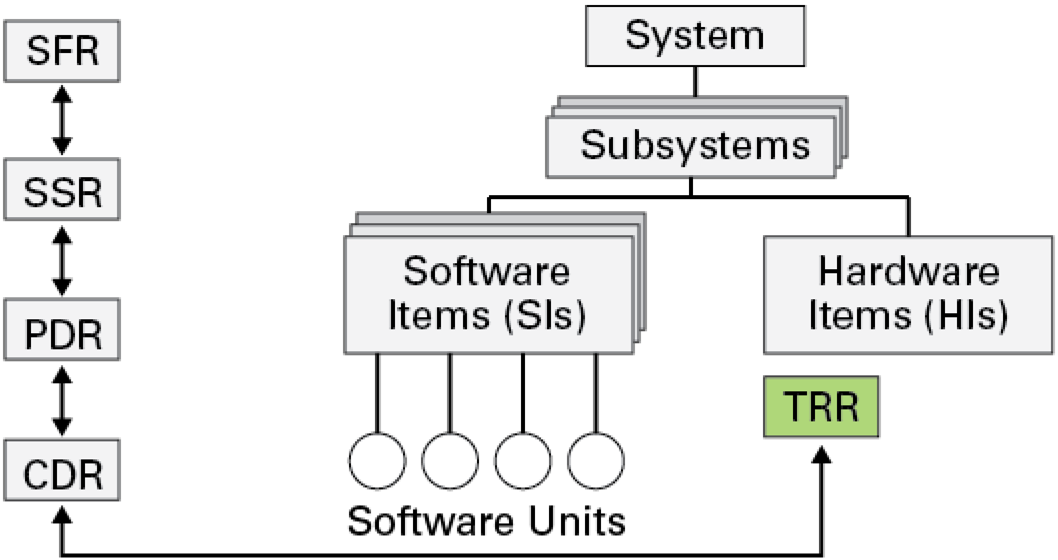


D

Test Readiness Review (TRR)

During the Test Readiness Review (TRR) reviewers seek to:

- Confirm completeness of proposed test procedures and compliance with plans
- Verify readiness for formal Qualification Testing
- Ensure that the performing activity is prepared for formal testing



D