

CH1:

Purpose of PROGRAM AND PROJECT PLANNING: to provide the necessary process steps to plan for systems and software design and development for integration.

Result of PROGRAM AND PROJECT PLANNING: This type of planning will ensure effective results for performing the disciplines necessary to implement supporting software and systems integration activities.

Purpose of SOFTWARE REQUIREMENTS: the basis for software design and/or development.

Purpose of SOFTWARE DESIGN/DEVELOPMENT: a systematic approach used to create software design and its development to reflect design and software definitions related to the work product.

Purpose of SOFTWARE IMPLEMENTATION: provides assurance that engineering builds function as expected.

Purpose of SOFTWARE AND SYSTEMS INTEGRATION: provides a consistent approach to integration to ensure that the software and systems elements are assembled properly.

SOFTWARE SUBCONTRACTOR Role: to describe how a programs and projects will benefit from outside resources.

Purpose of SOFTWARE AND SYSTEMS INTEGRATION DELIVERY: to ensure that units tested are complete and documented prior to official delivery.

Purpose of PRODUCT EVALUATION: provides the necessary process to conduct and perform continual evaluations of software products during the design/development life cycle and integration activities.

CH2:**Implement EFFECTIVE PROGRAMS:**

- Required data
- How the work product performs
- Tasks or functions
- Quantitative mechanisms

ESTABLISHED FRAMEWORK:

- Effective planning entails multiple tasks, scheduled milestones, and quality aspects for everyone involved from management to employees.
- Configuration management personnel monitor the framework process.

components of a quality project guideline:

- Daily meetings
- Idea sharing
- Keep project managers informed
- Complaint resolution

COMMUNICATION PLANNING PRINCIPALS:

- Define and understand quality
- Define goals and objectives
- **Establish a set of managers who:**
 - Understand the technical practices that support systems and software engineering
 - Can clearly define and provide a scope for the team defining the development stages
 - Provide a scope for the team to know what is ahead
 - Involve systems and software teams to help with delivery schedules
 - Can accommodate change and identify potential risks that impact on program and project planning
 - Track the progress daily and adjusting if needed

SENIOR MANAGEMENT Role: provide the common framework for program and project planning to address engineering tasks

SENIOR MANAGEMENT Responsibilities:

- Communicate efficiently and manage a team wisely
- Implement and use reasonable schedules
- Oversee the development of a quality work product that meets the needs of the customer
- Demand the best from designers and developers

PLANNING ACTIVITIES SHOULD INCLUDE:

- Lessons learned from previous programs and projects
- Cost/schedule estimates and plans for staffing
- Definitions for software and system requirement
- Requirements for safety and security
- Selection of software subcontractors
- Engineering documentation and historical data impacts
- Objectives for program/project
- Contract interpretation of necessary requirements

PLANNED SCHEDULES:

- **Purpose:** defines tasks and processes to be conducted for implementation
- **Importance:** planned schedules affect team capabilities for risk assessment, configuration control, and quality.

THE THREE CRITICAL FACTORS: Scope, Budget, Quality**CRITICAL ITEMS IN A DEVELOPMENT PLAN:**

- Planned schedules
- Engineering information
- Software production direction
- A process consistent with system-level planning
- Consistency in agreement with the steps outlined

DEVELOPMENT PLANNING STEPS:

- Define entry and exit criteria for the software design/development
- Review and assessment of the work product/task requirements
- Define/update the process for each software activity
- Develop/update the estimating process
- Develop initial cost with schedule estimation and potential risks included
- Prepare detailed implementation plans

THE IMPORTANCE OF TEAMWORK:

- Energy and consistency influence high-performance goals. Therefore, trust and cohesiveness must be maintained in the work environment.
- A plan is successful when a team delivers a high-quality work product, meets the defined schedule and maintains budget.

TEAM RESPONSIBILITIES:

- Meet and achieve team objectives
- Resolve conflicts and issues
- Satisfy customer requirements

TEAM ACTION CYCLE:

- DETERMINE GOALS
- CREATE RESPONSIBILITIES
- DEVELOP A PLAN
- TEST OUTCOMES
- COMMIT TO ACTION AND FOLLOW UP

CH3**EFFECTIVE TEAM MEMBERS:**

- Know how to manage their own reactivity
- Are curious about what caused the blame
- Let members of a team know how something has an impact on them
- Rely on others' experience and expertise

Purpose of SYSTEM DESIGN:

- to ensure an accurate and complete understanding of the restrictions of a system that affect work products.
- The external software interface is defined and verified for completeness.
- External interfaces based on software architecture definitions are identified as part of consequent software requirements.

Purpose of SYSTEM ENGINEERING PLAN: to develop software requirements, analyze the system architecture, design and allocates system requirements.

A system engineering plan (SEP): can be written to establish system-level technical reviews

MAJOR TECHNICAL REVIEWS & AUDITS AFFECTING SOFTWARE AND SYSTEMS INCLUDE:

- Initial requirements (IR)
- Incremental design review (IDR)
- Final design meeting (FDM)
- Test readiness (TR)
- First-article inspection (FAI)
- Functional configuration audit (FCA)
- Physical configuration audit (PCA)

Three main sections of Organization:

- Systems engineering
- Technical program processes
- Engineering integration

Importance of team:

- documents/provides the technical expertise to execute activities throughout a software design/development life cycle.
- enables performance to be more effective
- enables technical planners to spend more time planning to ensure the customer assurance and satisfaction in addressing the technical challenges

SOFTWARE ARCHITECTURE EVALUATION:

- **Purpose:** provides a common approach to developing the work product architecture.
- **Application:** implementation of enhancements for change or corrections to existing software architectures.
 - Provides the viability of software architecture definitions to be applied
 - Conflicts in requirements, architecture and project plans are reported to product teams for resolution
 - Program and project are analyzed to determine the impacts on architecture development.
- **Objectives:** operational scenarios and system or subsystem requirements
- **Scope:** does use interface requirements to analyze operational designs, software risks, and plans to determine the objectives of the architecture.
- **Development:** identified during development and made available to be understood before beginning a software design/development life cycle.

THE SYSTEM/SUBSYSTEM ARCHITECTURE REQUIREMENTS DETERMINE IMPACTS:

- Influences to quality
- Functional necessities for determination of the software architecture

PROCESS RESULTS:

- Trade-offs between quality performance and the modifications are prioritized, identified outside system or subsystem requirements reviewed
- Determination of requirements to be modified
- Exhibits how satisfactory the architecture meets objectives, constraints, and quality attributes.
- Determines appropriate design methods to ensure problems are addressed.

CH4

Purpose of DEFINED SOFTWARE REQUIREMENTS: provide a systematic approach for program and project development of software requirements delivered by a number of ideas and solutions.

ANALYSIS: a step-by-step process to develop requirements for software work products.

USE CASE: to describe a flow of operations for the performance of systems and software implementation.

ARCHITECTURE: Identified/ defined as part of derived software requirements.

Purpose of INTEGRATION: transforms functional architecture into optimal design solutions.

VERIFICATION AND VALIDATION: Verification and validation begins with the review of software requirements.

Requirement Priority Purpose: determines the extent of verification and validation for each defined requirement.

Risk Reduction Assessment: ensures software design/development activities satisfy user needs in a manner that is efficient and cost-effective for integration of validation and requirement verification.

REQUIREMENTS DOCUMENTATION: documented based upon development plans, software processes, and product standards.

Critical Requirement: defined and complete software requirements must be in place before formal review.

Software requirements tools should:

- Have ability to impose multiple format requirements
- Support traceability and impact analysis
- Support software baselines and releases
- Alert modifications of requirements database

CH5

Purpose of DEVELOPMENT PLAN: a well-defined process useful for implementation and applicable standards.

Purpose of SOFTWARE REQUIREMENTS EVALUATION: defines software operational scenarios to ensure problems affecting software design are identified, evaluated, and resolved.

Purpose of PEER REVIEWS: to find and correct as many errors as possible before test team integration or customers find problems.

PEER REVIEW METHODS:

- Inspections
- Structured walk-throughs
- Deliberate refactoring
- Pair programming

PEER REVIEWS CRITERIA:

- Schedule the peer review at a convenient time
- Assign reviewers
- Prepare/update materials
- Provide checklists
- Introduce training materials
- Select software work products
- Provide entry and exit criteria

Lean Software Design/Development:

- to move as many changes as possible from the top curve to the bottom curve.
- Delays the freezing of all design decisions as long as possible.
- Emphasizes designing and managing changes throughout the life cycle.
- Provides a better understanding of software engineering and quick delivery to customers.

AGILE SOFTWARE PROCESSES:

- Provides fewer defects.
- Supports numerous initiatives
- Provides a program and project with a manager's approach to emphasize short-term program and project planning.
- Adopts values that are consistently depicts processes and makes decisions that may reject a software design.
- More effective than the traditional models due to perfection versus good-enough concepts for software design practices.
- Provides capability to understand information first before jumping into software design and development.

KEY ELEMENTS OF AGILE SOFTWARE ENGINEERING:

1. The team has control of work assignments
2. Communication with team members and customers is needed
3. Change is good: "Think outside the box"
4. Customer satisfaction and expectations are achieved

CONFIGURATION MANAGEMENT methods: are a supporting discipline not directly involved in creating executable code. In the Agile process, CM methods are not referenced for specific routines.

Configuration Management Purpose: trim process and provide more automation in tools

Purpose of SOFTWARE STANDARDS: ensures development processes are in accordance with identified process models.

Minimum software standards consist of the following:

- Documented and maintained plans and procedures
- Peer reviews
- Standard software tools

Capability Maturity Model Integration ENGINEERING TASKS:

- Identify internal and external interfaces
- Establish infrastructure abilities with software design
- Develop plans, processes, and procedures
- Reuse capabilities for identified software

Purpose of LEAN SIX SIGMA: reduces process variation, resulting in fewer errors and defects. **And the Goal** is Zero defects.

Six Sigma Process:

- Define
- Measure
- Analyze
- Improve

LEAN GOAL: ELIMINATE EIGHT WASTES:

- | | | | |
|-------------------|-------------------|------------|-----------------------|
| 1. Defects | 2. Overproduction | 3. Waiting | 4. Nonutilized talent |
| 5. Transportation | 6. Inventory | 7. Motion | 8. Excess processing |

CH6

Team Focus: identifies and manages changes and maintains software configuration and documentation visibility.

Purpose of BUILD REQUESTS: provides checklists to assemble, compile, link source code, build archive copies, and provide listings for use in software design/development, test, and work product customer delivery.

BUILD ENGINEER ROLE:

- Creates build folders to store documentation of software building
- Provides source code changes and control of the source code
- Maintains and controls records during program and project development

Purpose of CONFIGURATION MANAGEMENT TOOLS: provides the capabilities for adding new files to a software design/development environment in addition to providing version control to directories and files.

Essential Elements of CONFIGURATION MANAGEMENT TOOLS: File sharing, parallel software design/development, multiple team support, and software reuse to ensure integration test activities demanded by the schedule.

IBM RATIONAL CLEARCASE: An object-oriented database utility provided to establish software product archiving, automation, identification, version/change control, engineering building, product releases, status accounting, and auditing activities.

Purpose of IBM RATIONAL CLEARCASE: provides an open architecture to implement configuration management and control solutions.

CLEARCASE ROLES:

- Architect
- Configuration manager
- Lead
- Software design engineer
- Build engineer

Purpose of IBM RATIONAL CLEARQUEST: provides support for change request management processes and is a ClearCase complementary tool.

Compare and contrast ClearCase and ClearQuest

<u>ClearCase (IBM)</u>	<u>ClearQuest (IBM)</u>
Design, code, and unit test, software builds/installation, integration, and test	Code and unit test, integration, and integration testing
Tools for documentation and source code, support multiple developments, and release baselines	Software problem reporting, logs, tracking, and software debugging and fixes

Media Label Documentation Items:

- **Date:** Day/month/year format
- **Title:** Document the title of the software being produced
- **Derived from:** Program and project
- **Special handling:** Distribution requirements
- **Contract number:** Document contract number
- **Part number:** Document software identifier
- **Software version:** Media version

FUTURE TRENDS: Improvements in software technology will address and resolve issues and improvements required for:

- Software design/development
- Software process definition and enhancements
- Reuse of software program and project artifacts
- Ongoing support of past tool artifacts
- Training for software design engineers
- Software tool disciplines

What are the primary steps of tool support?

- Become effective for designing and developing work products
- Establish the resources for use of software tools
- Conduct software implementation with no problems
- Conduct training

CH7

Purpose OF SOFTWARE INTEGRATION STRATEGY: describes the steps to be conducted as part of the implementation of software to start integration activities.

Importance OF SOFTWARE INTEGRATION STRATEGY: flexibility with an approach that to show change.

PLANNING CHARACTERISTICS:

- Effectively conducted technical reviews
- Differentiated integration techniques and software approaches
- Required involvement of software designers from start to finish

Approach to Software Integration: Planned in advance as the start for effective software integration.

- accommodates lower-level integration to verify software code development that has been implemented correctly and major system functional expectations met.
- provides guidance software design/development and test teams to reach milestone expectations.
- The steps occur many times each time deadlines occur.
- Measurement problems resolved early in schedules.

What is software integration testing?

The concept for testing software is to uncover errors, troubleshoot, and fix problems that occur during a test.

bam theory approach: is to attempt nonscheduled software integration and testing.

Step for bam theory approach:

1. Software test plans, procedures, or internal work instructions are ready to support integration
2. Software integration is ready for testing to be conducted and performed by all team members
3. Control must be maintained between multiple tests running at the same time (lack of control can cause chaos)

Use of DEVELOPMENT FACILITY: to prepare software prior to delivery to a software systems integration facility (S/SIF).

What is the Intrinsic adaptation for software operations include ?

- parameter-based initialization data
- settings selected/entered by designer/developer
- test teams during operations of the software
- systems retained for other test integration purposes.

Configured Baseline Purpose: identifies the development life cycle- specifically, functional and allocated work product baselines, **Unique software documentation and media** define software configuration baselines.

SOFTWARE INTEGRATION SETUP: Involves planning with program and project managers to coordinate with the facility operations manager.

Allocated resources: (i.e., computers, workstations, and hardware units) provided to the software designer/developer and test teams to conduct informal integration testing.

Define the Installation Plans and its Procedures ?

- **Software integration test plans:** covers the testing of requirements and verification methods conducted in the DF.
- **Specific integration test plans:** consist of checkout activities to ensure system utilization.
- **Integration testing environment:** provides necessary steps to be followed, and analysis solutions are used
- **Installation test plans:** peer reviewed and approved for release by program and project managers

SOFTWARE INTEGRATION LOG: provides a view of the day-to-day operations for the design and test teams using hardware units for integration and checkout.

Verification: set of tasks to ensure correct implementation techniques are in place and product integrated correctly.

Validation: ensures the correct work product is the correct product to validate.

QUALITY TEAM ROLE:

- Performs technical reviews
- Audits configuration management
- Software integration progression monitoring
- Reviews plans, procedures, and documentation
- Tests qualification and acceptance
- Witnesses implemented plans/procedures during integration/testing

What is importance of configuration reviews and Audits? ensures all elements of software configurations are developed and in control during software integration and test activities.

CH8**Successful software and systems integration objectives are accomplished by:**

- Agreeing on and identifying blocking issues
- Assigning responsibility for clearing those blocking issues
- Scheduling dates for responsible teams
- Holding periodic meetings for issues and concerns
- Evaluating current integration facility schedules

Purpose of software and systems integration plan (SSIP): defines processes/procedures used to integrate defined work products, systems or subsystems, and hardware units into a software/systems integration.

Facility Operations Purpose: determines software design/development and tests to be conducted and establishes the environment for software and hardware configurations.

Facility Configuration Purpose: to support design and test operations.

Responsibility of Test Team: The test team is responsible for formal qualifications of a specified system requirement.

What are the required documentation of test team?

- SSIP
- Integration and installation procedures
- Design documentation
- User and operation guides
- Test and analysis reports
- Compliance documentation or sheets
- Hardware drawings

Purpose of INTEGRATION TEST PROCESSES: ensure acceptance testing has been completed at the end of formal qualification testing.

Integration Test Processes:

1. Develop Test
2. Develop Procedures
3. Acceptance Test

The quality checklist will provide:

- Criteria defined from previous audits, plans, procedures, and documented requirements
- Recorded results, including any noncompliance or observations
- An audit report that provides the scope and purpose of the audit, completed checklists, trained personnel, results and lessons learned for future improvements
- Measurement data produced during the audit
- Applicable work products submitted for control in accordance with the software/system plans

Effective Methods for Software and Systems Integration:

- Planning
- Systems/software design
- Configuration management
- Communication
- Integration
- Quality
- Risk management
- Execution
- Customer satisfaction
- Requirements
- Continuous integration

basic process steps of risk?

- Risk issues and concerns.
- Risk reviews.
- Risk management plans.
- Risk monitoring.

Risk-Based Integration: reviewed when analysis is performed to root out software design/development and test defects.

Evidence of Requirements: program- and project-developed software and commercial off-the-shelf (COTS) or non-development items (NDIs) elements are defined and documented.

COMPONENTS OF SUCCESS:

- **SYSTEMS/SOFTWARE DESIGN:** definition developed and controlled by plans for development/ design.
- **INTEGRATION:** implements a readiness review to ensure the lab environment is ready for design and testing.
- **EXECUTION:** software/system integration show execution of test-built systems for integration activities and ensure the builds are not broken.
- **CONFIGURATION MANAGEMENT:** a discipline applying an administrative process in addition to direction for work products developed.
- **QUALITY:** essential practice for successful teams to follow.
- **CUSTOMER SATISFACTION:** assuring the customer that effective methods for software and systems integration have been compliant and meet concrete requirement expectations.

CH9

Purpose of Software Subcontractor: provide the necessary support and employment that would benefit for program and project planning, configuration management, quality issues, software design/development, testing, and execution of activities/tasks related to the delivery of software work products to customers.

Why The selection of a subcontractor for software programs and projects is important?

Because of expectations and technical work disciplines required.

Statement of work (SOW): will list subcontractor requirements.

Purpose of SOFTWARE SUBCONTRACTOR PLAN: provides direction for the subcontractor, program, and projects for understanding the requirements and guidelines.

Subcontractor Responsibility: configuration management software at the subcontractor's facilities in accordance with the plans/procedures while abiding to the standards, processes, and procedures of the program/projects under a signed contract.

What the Subcontractor plan approach describe?

- Subcontractor task
- Configuration management and quality audits process
- Job requirements
- Management of risk
- Management configuration
- Delivery schedules for work products

subcontractor process Software audit criteria: prepared/provided to subcontractor before audits are performed.

- audit checklist is provided, audit questions will be filled out, presented to the customer.
- agenda and participants are identified with a defined audit process applicable per the contract.

CH10

IMPORTANT ASPECTS BEFORE DELIVERY OF SOFTWARE AND SYSTEMS TO CUSTOMERS:

- Completion of software media and data verification and validation.
- Release of software documentation and readiness for delivery.
- Conduction of necessary FAIs (first-article inspections), FCAs (functional configuration audits), and PCAs (physical configuration audits).
- Closure of all action items.

SOFTWARE MEDIA AND DATA DELIVERY:

- Must be in accordance with security requirements for a program/ project per defined and documented security plan.
- Marking information displayed electronically for software media and on the exterior of the physical media (i.e., disk sets, DVDs, CDs, etc.) containing software.
- Software work products are identified in program and project development plans.
- Identification approach assigned to released software and accompanying software documentation.

Software Documentation list:

- Systems engineering plan (SEP)
- Development plan (DP)
- Software configuration management plan (SCMP)
- Software test plans and procedures
- Software and systems integration plan (SSIP)
- Quality plan (QP)
- Documentation for version control
- Build and installation procedure

Version Control Documentation: provides version control to identify and describe software versions of existing work products, **Use:** release, track, and control software versions at software and system levels.

Purpose of build and installation procedure: describes in detail how to build and install software for systems integration and it is configuration management team Responsibility.

Software deliveries Package: are used to meet contract delivery requirements or obligations the program has agreed to accomplish.

What is Final Software and Systems Delivery? is the last delivery once program and projects have completed the FAI and FCA/PCA.

Purpose of FIRST ARTICLE INSPECTION(FAI): examines subcontractor production units and determines whether the software is ready for delivery to the customer.

SOFTWARE First Article Inspection (FAI) CHECKLISTS:

- Verification requirements
- Data package
- Version control document
- Verification process
- Product release
- Acceptance test
- FAI completion

Purpose of FUNCTIONAL CONFIGURATION AUDIT(FCA): verifies work product performance complies with the hardware, software, and interface requirements specification (IRS), to provide the prerequisite to acceptance of a configuration item.

Functional Configuration Audit (FCA) Activities:

- Verifies work product performs to required configurations.
- Changes release for major or minor engineering
- Establishes baseline for product and configuration

Purpose of PHYSICAL CONFIGURATION AUDIT (PCA): identifies the baseline for production and acceptance of the work product both hardware and software.

PHYSICAL CONFIGURATION AUDIT (PCA)ACTIVITIES:

- As-built configuration in correlation with the as-designed product configuration.
- Determine acceptance test requirements in accordance with quality assurance.
- Release engineering changes.
- Establish final product baseline.

CH11

What is PRODUCT EVALUATION? integral part of program/project-level activities, scheduled and performed by quality software personnel on an ongoing basis.

Purpose of PRODUCT EVALUATION: form the basis for certification that software design/development activities were performed in accordance with program and project plans/procedures and are in line with required quality requirements.

Purpose of QUALITY ASSURANCE: provides product evaluation processes and specific quality assurance for effective software engineering methods and software tool use.

Purpose of software quality plan (SQP): describes/documents software quality assurance roles/responsibilities to ensure that programs and projects are following procedures and processes defined per development plans and other applicable standards.

Purpose of Quality reviews/internal audits are performed to ensure compliance with released processes and AS9100C for measurement, analysis, and improvement activities to be conducted.

Quality reviews/internal audits activities:

- Program and project artifacts
- Released processes and procedures
- Configuration management requirements of Electronic Industries Alliance (EIA) standards organizations are met
- Requirements of AS9100C are met by organizations

program and project managers are responsible to issues relating to:

- Team objectives
- Risk mitigation
- Issues and concerns
- Root cause (RC) analysis
- Corrective action (CA) plans
- Significant accomplishments

The focus of successful program and project managers is: team, processes, and work product.

Components of Inclusion:

- software configuration records
- testing records
- Audit records
- Audit and product evaluation checklists
- Audit results and audit reports

Product evaluations include:

- Review of plans and procedures that oversee programs and projects to determine and select appropriate product evaluation criteria
- Review and analysis of the results of previous product evaluations
- An assessment of whether implemented processes are compliant or noncompliant
- Identification of issues or an opportunity for improvement
- Additional product evaluations required

Purpose of Corrective Action: is required to eliminate or mitigate the cause of a detected nonconformity or other undesirable situations to prevent recurrences during product evaluation.

Corrective Action Type:

- **The root cause (RC):** requires RC analysis and actions taken to address the analysis.
- **The immediate action (IA):** taken to address a direct cause and prevent recurrence of a specific nonconformity.

Purpose of SOFTWARE REVIEWS: provide the framework and detailed requirements for verifying/validating design/development efforts.

SEVEN STEP STRESS MANAGEMENT TECHNIQUE:

1. Picture yourself near the ocean.
2. The ocean is blue and crystal clear.
3. Birds are flying by and chirping.
4. You are the only one there and in total seclusion.
5. There are soothing sounds, and the air is filled with serenity.
6. You can easily make out the faces of the team members under water.
7. See—you are smiling.

Master Schedule Purpose: defines the dates that serve as milestones for software/systems integration activities.

SOFTWARE TOOLS: equipment, location, and tools used to develop and test facility environments.

CH12

Software plans and procedures include:

- Development plan (DP)
- Software user manuals (SUMs)
- Software configuration management plan (SCMP)
- Software build and installation procedures
- Software quality plan (SQP)
- Integration facilities operational plans and procedures

SOLVING QUALITY ISSUES: Quality engineers:

- Solve process issues and concerns they are unable to fix code and test software but participate in peer reviews and witness integration test activities.
- Provide assistance and help program and project managers
- They must listen to and understand the roles and responsibilities of quality engineering.
- Quality factors are essential and important to understand.

System design plans and procedures include:

- Systems engineering plan (SEP)
- Systems design procedures
- Systems test reports
- Hardware drawings
- Integration facility configurations
- Hardware serial number verification procedures or instructions

Hardware includes:

- Workstations
- Display units
- Printers
- Disk sets
- Drivers and interface cards
- Networks or servers

Test documentation includes:

- Test plan (TP)
- High-level test plans
- Test report (TR)
- Installation procedures

Informal Test: based on informal and draft documents, not released to customers -maintained internally for checkout, troubleshooting, and recommendations to start formal test.

Formal Test: states objective and success criteria, requirement describes the configuration of software/system item under test and lists test equipment and required support.

Responsibilities of Systems Engineering: allocate technical requirements for programs/ projects that consist of both software and hardware.

Responsibilities of Software Development: responsible for troubleshooting, resolving software problems, and supporting isolation of system problems occurring while testing hardware and software.

Responsibilities of Software Configuration Management: responsible for building and providing configuration control of software for test organizations to use in the DL and integration facility.

Responsibilities of Software Quality Organization: support system and software tests conducted as defined in the software quality plan (SQP).

Verification and validation is important part of software requirement. Justify this argument.

- Reduces the risk
- Ensures design and development satisfies the user need
- This ensures the what is understood by analyst is exactly what is required by the client.

What is the difference between the following software design/development methods (state the definition and advantages/disadvantages):

- **Concurrent Software Design/Development method**
- **Lean Software Design/Development method**

Concurrent Software Design/Development method:

- **Requirement:** software design expertise to anticipate where the defined design is going.
- **Disadvantage:** possible to delay commitment until the last moment when failure to make a decision eliminates an important alternative or decision.

Lean Software Design/Development method:

- **Objective:** to move as many changes as possible from the top curve to the bottom curve.
- **Advantages:** Delays the freezing of all design decisions as long as possible.
- Emphasizes designing and managing changes throughout the life cycle.
- Provides a better understanding of software engineering and quick delivery to customers.

تم بحمد الله دعواتكم