

CHAPTER 6: SOFTWARE IMPLEMENTATION

SOFTWARE IMPLEMENTATION METHOD

Purpose: provides assurance that software engineering builds function as expected and enables smooth execution for verification and validation activities.

- The necessary requirements for software implementation is released under configuration management control as described in a defined documented configuration management plan (CMP).

CONFIGURATION MANAGEMENT

Purpose: ensures configuration management practices are applied consistently throughout the software life cycle for programs/products.

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

Processes: controls storage, access, changes, archive, and release of the software work products.

Procedures: describes implementation of processes required to meet requirements and direction provided under plan association and documentation.

BUILD REQUESTS

Purpose: 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.

Processes: include the capability to package builds and documentation together.

- Requires coordinated communication between internal and external teams to be efficient and available for scheduled tests or configuration checkouts.

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

CONFIGURATION MANAGEMENT TOOLS

Purpose: 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: File sharing, parallel software design/development, multiple team support, and software reuse to ensure integration test activities demanded by the schedule.

IBM RATIONAL CLEARCASE

Definition: 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: provides an open architecture to implement configuration management and control solutions.

CLEARCASE ROLE

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

IBM RATIONAL CLEARQUEST

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

Use: recording, tracking, and reporting and provides internal access control mechanisms for permitting the restriction of work product updates throughout the various stages of software design/development, integration and test, and production processes.

SOFTWARE MEDIA AND DATA

Physical software media identification and labeling must be in accordance with program and project documented media requirements.

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

TOOL SUPPORT

Key: selecting the right vendor products to match engineering needs.

Questions: necessary for meeting organizational needs.

PRIMARY STEPS

1. Become effective for designing and developing work products
2. Establish the resources for use of software tools
3. Conduct software implementation with no problems
4. Conduct training

CHAPTER 7: SOFTWARE INTEGRATION

SOFTWARE INTEGRATION

Methods: provide required steps to be conducted for integration and checkout of informal software engineering builds.

Activities: informal and flexible for software checkout to prepare for the software and systems integration phase.

SOFTWARE INTEGRATION STRATEGY

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

Importance: 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 INEGRATION

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 numerous times each time deadlines occur.
- Measurement problems resolved early in schedules.

SOFTWARE INTEGRATION TESTING

Concept: to uncover errors, troubleshoot, and fix problems that occur during a test.

Purpose: having plans and procedures in place ensure that testing strategies are not wasted time during integration.

“BAM THEORY” APPROACH

Purpose: to attempt nonscheduled software integration and testing.

Three step approaches:

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 notified team members
3. Control must be maintained between multiple tests running at the same time (lack of control can cause chaos)

DEVELOPMENT FACILITY

Purpose: established early on in software design/development phases for software integration activities.

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

Overview: includes geographic locations, facilities used, and secure areas.

Requirement: equipment furnished by customer, software, services, documentation, data, facilities, and a detailed schedule highlighting when needed items are included.

- Additional required resources: plans for obtaining the resources, dates needed, and availability of each resource item.

SOFTWARE OPERATIONS

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.

SOFTWARE DESIGN/DEVELOPMENT ENVIRONMENT REQUIREMENTS

- Plans must ensure each element of an environment performs to intended functions.
- Plans provide requirements for test environments to perform software testing, including integration, troubleshooting, and checkout
 - ensure each element of the test environment performs intended functions.

SOFTWARE CONFIGURATION

Software Configuration Identifications Purpose: ensures configuration control.

Configured Baseline Purpose: identifies the development life cycle- specifically, functional and allocated work product baselines.

- Configured baselines are defined by unique software documentation and media software.

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.
- Software engineering builds and loading into hardware units are performed by build engineers.

INTEGRATION TEST

System integration tests are conducted inside programs and project integration facilities.

- Includes verification steps: ensures tests provide a check of the capabilities of software and hardware units.
- Repeated numerous times: ensures all integration test problems are resolved, performance is accomplished early in the defined system, and the system is working to software requirements.

INSTALLATIONS PLANS AND PROCEDURE

Purpose: defines systems' specification requirements.

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, data collected, and analysis solutions are used or implemented

Installation test plans: peer reviewed and approved for release by program and project managers

- prepares for the start of software integration testing.

SOFTWARE INTEGRATION LOG

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

Use: supports operational setup activities.

- Software design teams have the ability to fix/debug problems and work with test teams
- Ensures plans and procedures will be ready for release to support formal test phases

INTEGRATION VERIFICATION AND VALIDATION

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

CONFIGURATION REVIEWS AND AUDITS

Purpose: ensures all elements of software configurations are developed and in control during software integration and test activities.

Key Timing: before entering into formal software and systems integration.

CHAPTER 8: SOFTWARE AND SYSTEMS INTEGRATION

SOFTWARE AND SYSTEMS INTEGRATION OBJECTIVES ARE ACCOMPLISHED THROUGH

- 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

SOFTWARE AND SYSTEMS INTEGRATION PLAN

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

Includes: software integration planning in coordination with other formal test activities, risk assessment, product evaluations, configuration management (CM), and any other necessary support activities.

SOFTWARE AND SYSTEMS INTEGRATION FACILITY

Purpose: primary facility for hardware, software integration, and system-level testing.

Supports: software design and hardware equipment integration

FACILITY OPERATIONS AND CONFIGURATION

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

- Ensure systems integration facility operations are conducted, systems are integrated, and performance is measured.

Configuration Purpose: to support design and test operations.

- Documented drawings to lay out facility configuration and coordinate with hardware, electrical engineering, and hardware quality.

INTEGRATION SETUP

- Occurs within the system, emphasizing interfaces and operations between components, such as hardware/software, interfaces, and other supporting functions.
- Work products integrated and performed incrementally.

TEST TEAM

Responsibility: formal qualifications of a specified system requirement.

Work Environment: inside the facilities' operations with other systems and software personnel.

Required Documentation: defines and documents the progression and interdependency of test artifacts.

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

INTEGRATION TEST PROCESSES

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

Process:

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

QUALITY PARTICIPATION IN SOFTWARE AND SYSTEMS INTEGRATION

Purpose: ensures software and systems hardware work as one.

- test team runs through test installation procedures with the quality team and results are documented for completion and closed.

Common Approach: redlines applied to installation procedure are authorized and for the next formal release procedure to support testing.

QUALITY CHECKLIST

Provides:

- Criteria defined from previous audits, plans, procedures, and documented requirements
- Recorded results, including any noncompliances 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

VERIFICATION AND VALIDATION

Purpose: addresses work products in integration environments and includes selected requirements

- incremental process performed throughout the software design/development life cycle.

Validation Process: ensures compliance to plans, procedures, and data inside integration facilities.

LATE NIGHTS, EARLY MORNINGS, AND WEEKENDS

Software quality team is required to support software and systems integration activities in the S/SIF at various times and must be available.

- **Buy-off Plan Requirement:** quality team verification and validation and approvals applied

SOFTWARE QUALITY SUPPORT

Efficiency is important for execution of quality tasks.

Tasks Include: process/product evaluations, reviews, audits, planning, formal audits, training, and verification and validation of work products to be ready for formal test and delivery.

EFFECTIVE METHODS FOR SOFTWARE AND SYSTEMS INTEGRATION INCLUDE

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

PLANNING

- Develop the SSIP and strategy to understand the systems being integrated.
 - Include the environment, functions, and constraints
- Ensure requirements are testable, operational, and technically realistic.
 - Consider using an integration readiness review plan
- Monitor Planning Progress
- Communicate: tell senior, program, and project managers to be honest with teams and the customer

RISK MANAGEMENT

- Conducted for integration of software and systems: must continuous and shows the risks that occur.
- Risks must be documented and reviewed each day.
- Continuous process of identification to resolve problems.

BASIC PROCESS STEPS

- 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.

Risk Integration Standards: identified in ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) standard 61508.

REQUIREMENTS

- Define and develop software requirements selected for implementation and completion during software and systems integration.
- Software requirements identified for automation of builds and installations inside the software and systems integration environment.
- Software work products integrated correctly and reflect continuous improvement.

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.

CHAPTER 9: SOFTWARE SUBCONTRACTOR

SOFTWARE SUBCONTRACTOR

Purpose: 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.

Activities: in accordance with a purchase contract, and the software work products are delivered to satisfy and comply with specified requirements.

PROGRAM AND PROJECT SELECTION

Importance: selection of a subcontractor is important because of expectations and technical work disciplines required.

- Hiring-responsibility of program and project managers.

- Provide direction to perform job tasks related to the day-to-day software design/development to be delivered for subcontracted work products.

Planning: performed during program and project start-up

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

Search and Selection of Subcontractor: required by program/project plans or the subcontractor program or project managers.

- Support includes teams or coordinated experience of software engineers for understanding all aspects.

SUBCONTRACTOR APPROACH

- Software subcontractor is an integral part of the team for product development.
 - Ensure progress by the appropriate subcontractor
- Software activities/progress based on agreed evidence of completion.
 - Responsibility of program and project organizations as well as senior managers to oversee the subcontractor's work quality

SOFTWARE SUBCONTRACTOR PLAN

Purpose: 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.

SUBCONTRACTOR PLAN APPROACH

The plan should describe the following in detail:

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

SOFTWARE AUDITS

Subcontractor software plans and procedures are audited per defined and documented audit methods to trace information for software requirements per the signed contract.

- ensures software test environment performs intended functions and meets contract requirements.

Purpose: ensures that the software under test is qualified on acceptable test tools.

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.

Approach: involvement in the software first-article inspection (FAI)

CHAPTER 10: SOFTWARE AND SYSTEM DELIVERY

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

Purpose: provides defined/documented releases for varying levels of software and systems integration.

Use:

- 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

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.

BUILD AND INSTALLATION PROCEDURE

Purpose: details description of how to build and install software for systems integration.

Responsibility: configuration management team, with input from software designers, procedures for software and systems integration builds.

- The CM organization is responsible for the development, control, and release of build and installation procedures.

DELIVERY PACKAGE

Consists of software media and documentation associated with the version of the software, printed copies, and identified system work products or hardware packages.

Purpose: used to meet contract delivery requirements program agreed to accomplish.

Responsible Parties: senior management and program/project managers in addition to identified teams.

FINAL SOFTWARE AND SYSTEMS DELIVERY

This is the last delivery after program and projects have completed the FAI and FCA/PCA.

Purpose: provides processes and procedures to ensure all aspects are correctly accomplished.

Result: programs and projects are able to sustain continuous improvement.

Risks: ineffective delivery systems.

Global Market Survival: programs and projects must continuously improve their work products, services, and delivery systems.

Programs and Projects: create work products and services to meet the needs of the customer.

FIRST ARTICLE INSPECTION

Purpose: examines subcontractor production units and determines whether the software is ready for delivery to the customer.

Subcontractors: work with customer's engineering teams to finalize regression analysis.

SOFTWARE FAI CHECKLIST

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

FUNCTIONAL CONFIGURATION AUDIT

Purpose: verifies work product performance complies with the hardware, software, and interface requirements specification (IRS).

Requirement: test data must be reviewed and verified in addition to showing that the hardware and software perform as required by configuration identification.

- Must have a technical understanding to accomplish the concerning the validation/ verification per the TP concerning software,

Use: to provide the prerequisite to acceptance of a configuration item.

FCA ACTIVITIES

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

PHYSICAL CONFIGURATION AUDIT

Purpose: identifies the baseline for production and acceptance of the work product configuration

- Ensures acceptance test requirements are comprehensive and meet the necessary requirements for acceptance.
- Demonstrates management systems for quality, engineering, and configuration management information accurately control configuration of subsequent production units.

Option: current conduction with the FCA.

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.

CHAPTER 11: PRODUCT EVALUATION

PRODUCT EVALUATION

Definition: integral part of program/project-level activities, scheduled and performed by quality software personnel on an ongoing basis.

Purpose: 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.

QUALITY ASSURANCE

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

Team Responsibility: to ensure compliance to software design/development standards and control work products.

Application: throughout the software design/development processes.

Management: summarized in engineering reviews, change control, or subcontractor audits and compliance to standards, verification, and validation.

SOFTWARE QUALITY PLAN

Purpose: describes/documents software quality assurance roles/responsibilities

- ensures programs and projects are following procedures and processes
- provides a documented process for assessing software life-cycle processes and their outputs

Audits: ensure compliance with released processes and AS9100C for measurement, analysis, and improvement activities to be conducted

AUDIT EXAMINATION INCLUSIONS

- 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

PRODUCT EVALUATION SCHEDULE

Purpose: performs product evaluations and ensures software design/development.

Test and integration phases: conducted according to a product evaluation schedule.

PROGRAM AND PROJECT MANAGERS' VISION STATEMENT

Definition: an example list that program and project managers are responsible for producing.

Includes: issues relating to team objectives, risk mitigation issues and concerns, root cause (RC) analysis, Corrective Action (CA) plans, and significant accomplishments.

Focus: team, processes, and work product.

ARTIFACTS

Include: software configuration records, testing records, and other artifacts associated with activities.

Additional Components of Inclusion:

- Audit records (i.e., electronic or paper) associated with product evaluations
- Audit and product evaluation checklists
- Audit results and audit reports

AUDIT FINDINGS

Purpose: utilize criteria audit finding derived from software plans and internal procedures to perform scheduled product evaluations.

Product Evaluations: include a review of plans and procedures that oversee programs and projects, a review and analysis of the results of previous product evaluations, an assessment of whether implemented processes are compliant, clearly defined issues or opportunities for improvement, and any additional product evaluations that are required.

Results: recorded in evaluation plans and added into databases recording summary information from evaluation performance.

Quality Team Use: to indicate if processes are compliant, noncompliant, or if there is an opportunity for improvement.

CORRECTIVE ACTIONS

Purpose: to eliminate the cause of a detected nonconformity or other undesirable situations and prevent recurrences during product evaluation.

Two Types of Initiated CA:

- 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.

APPROPRIATE REVIEWS OF CA AND IA

Monitor: progress/status and overall process performance (e.g., timeliness, efficiency, effectiveness)

Review: adverse indicators and trends

Resolve: issues or elevate them to the proper forum or level of management for resolution

QUALITY METRICS

Purpose: ensures program and project's delivery schedules, what is in-work, and completed product evaluations.

Achievement Requirements: effective methods and current software tools must be used.

Measurement Responsibility: trained senior managers and program and project managers.

QUALITY MANAGEMENT SYSTEM

Required to have processes documented and executed with knowledgeable people and teams

Purpose: provides the framework for philosophy of “do what you say, prove it, and show improvement”

Standards: AS9100, AS9100C, AS9100D, SAE AS9110, and ISO 900

Support Basis: conducting product evaluations, reviews, and audits for compliance to requirements insurance.

SOFTWARE PROCESS

Purpose: enforces process needs and follows these processes when product evaluations are conducted

Assessment: ensures process meets a set of basic criteria to show successful software engineering practices

- Leads to effective software and systems integration to improve processes

SOFTWARE REVIEWS

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

Process Improvement: entails communication, understanding, discipline and deployment.

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.

SOLVING QUALITY ISSUES

When problems arise with the delivery of work products to software and systems integration facilities and customers:

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.

APPENDIX B: SOFTWARE/SYSTEMS INTEGRATION PLAN

PLAN INFORMATION

Plan Type	Revised New Release Date	Contract Number
Formal/revision	dd/mm/yyyy	TBD or N/A

Signatures:

Author: Signature Program/Project mm/dd/yyyy

Check by: Signature Program/Project mm/dd/yyyy

Approved by: Signature Program/Project mm/dd/yyyy

Released by: Signature Program/Project mm/dd/yyyy

ABSTRACT

Purpose: defining plans, processes, and procedures for software/systems work product-level test and evaluation.

Components: computing hardware and software.

Software: an operating system and application.

Hardware: computers, displays, network interfaces, and interfaces to other subsystems.

SSIP: describes the test environment for testing, identifies tests to be performed, and provides an overview for test activities.

KEYWORDS

- Development File Folder (DFF)
- Development Lab (DL)
- Development Plan (DP)
- Quality Assurance (QA)
- Systems Integration Facility (SIF)
- Software Configuration Management (SCM)
- Software Engineering Institute (SEI)
- Software Systems Integration Plan (SSIP)

SSIP PLAN OVERVIEW

Definition: document for defining plans, processes, and procedures for the integration of software and systems for high-level developmental testing for programs/projects.

Processes Used: integration definition and development, integration procedure development reviews, integration activity execution, and integration evaluation and reporting.

Includes: overall integration planning and coordination and other test activities, risk assessment, product evaluation, software configuration management, and additional related software support activities.

Revision: as necessary throughout the *program or project* software development life cycle.

SENIOR MANAGEMENT FUNCTION

Purpose: performs guidance and support, monitor and report progress regarding software and systems integration, and define the role/task for software and systems activities.

Schedules: follows the program/project master schedule.

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

SOFTWARE TOOLS

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

Purpose: preserves integrity and structure of current software development and appropriate documentation.

Example: *ClearCase and ClearQuest.*

TEST

Purpose: provides verification to ensure that programs/projects fulfill requirements of software and system design.

Test Products: support software and system integration.

- Include: Software requirements, approved change, acquired engineering products, software resources, and controlled software authorization for updates.

SOFTWARE

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.

SYSTEMS

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

Hardware includes:

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

TEST DOCUMENTAION

Test documentation includes:

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

TEST APPROACH

Purpose: begins preparation of informal and formal test plans and procedures based upon listed specifications.

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

Requirement: require program/ project support during software and systems integration activities.

- **Systems Engineering:** allocate technical requirements for programs/ projects that consist of both software and hardware.
- **Software Development:** responsible for troubleshooting, resolving software problems, and supporting isolation of system problems occurring while testing hardware and software.
- **Software Configuration Management:** responsible for building and providing configuration control of software for test organizations to use in the DL and integration facility.
- **Software Quality Organization:** support system and software tests conducted as defined in the software quality plan (SQP).

FACILITIES OPERATION

Purpose: provides integration for hardware, software and systems integration to support software design and equipment integration.

Use: to integrate and test the work products and build/ support incremental deliveries of software builds.

Metrics: used on the *program/project* to manage facility operations activities.

- Used to evaluate the maturity of the software, measure progress of development, test efforts, and identify software risks during integration in an integration environment.

Risk Management: describes the process used by program/ projects to identify and monitor facility operation risks inside software and system integration activities.