Romel Cabiling

Home

Home > My courses > System Integration And Architecture 2 > 17 Systems And Software Development Process Summary > Lesson Proper for Week 17

Lesson Proper for Week 17

SOFTWARE AND SYSTEMS INTEGRATION METHODS

To develop, operate, and maintain software and systems integration capabilities inside work product facilities, there must be a major discipline in supporting the entire software life cycle (i.e., planning, systems, requirements, design, builds, installations, integration, subcontractors, quality, and delivery) that does need to be completely understood. The critical understanding and the start of the right disciplines of these methods will empower and achieve effective, flexible, and quality results in an integration environment.

PROGRAM AND PROJECT PLANNING

The purpose of program and project planning is to provide the necessary process steps to scope out planning for systems and software design/development within integration efforts. This type of planning will ensure and establish effective plans and results for performing the disciplines for software design/development, processes, and procedures for the implementation supporting software and systems integration activities. The planning and engineering task explains the disciplines and methods pertaining to communication, planning, risk management, and deployment.

Program and project planning is important as it describes the necessary planning for software and system efforts during software design/development life cycles. The definitions of systems design, software requirements and design, configuration control, systems and software integration, subcontractor involvement, deliveries, and product quality evaluations are critical to effective planning efforts. The initiation of planning starts at the proposal phase with the customer. The result of defined software design/ development plans, processes, procedures, subcontractor support, and effective software tools provides estimations for cost and schedules to be available for teams that are impacted from the start of the proposal phase to delivery of the work products to the customer.

SYSTEMS DESIGN

The method for systems design is to analyze customer requirements and develop a software design/development migration plan for defining the architecture, components, modules, interfaces, and necessary data for a designed system to satisfy specified requirements. The systems design method is increasingly important as it provides the disciplines required and implemented during software design/development life cycles.

The system/subsystem requirements reviewed by program and project personnel ensure accurate and complete understanding of the restrictions of systems design and applied work products. If program or project plans include reusable software interfaces; requirements are identified and evaluated for use. External software interfaces are defined as part of derived software requirements. To support systems design, graphical representations are prepared and take the form of data flow, collaboration/communications, and component diagrams.

SOFTWARE REQUIREMENTS

Defined and documented software requirements provide a systematic approach to development from multiple resources. The results of functional software interfaces, performance, verification, and production with required plans, documentation, and procedures become the basis for software design or development. This effective method is applied for initial development of software requirements and changes to requirement baselines.

Defined software requirements provide programs and projects with a systematic approach to the development of software requirements provided by various ideas and solutions. Software requirements establish the principals for software design and integration test activities for both software and systems integration. The generation and execution of software requirements are created as a stand-alone item or as an item embedded in higher-level assemblies (i.e., hardware units, workstations, monitor displays, integrated platforms, etc.).

SOFTWARE DESIGN/DEVELOPMENT

The software design/development definition is that of a systematic approach for the creation of software design and its development to reflect design and software definitions applicable to the work product. The resulting method defines details about the work product architecture, behavior, components, and interfaces. The software requirements are established between the elements of the design/development. The documented program and project plan provide traceability according to software-defined processes and procedures.

Software design is a consistent approach and method for the development of software requirements in defined designs of a work product. The software architecture definition provides a framework for the creation of the product design and at times can provide constrictions. The software design definition implements details about a software product’s architecture, components, and interfaces. Element traceability of the design and the software requirements is used by software designers. The traceability data and software design definitions are documented according to program and project plans, ideas, processes, and procedures and applicable internal work instructions.

SOFTWARE IMPLEMENTATION

The importance of software implementation is a requirement for informal and formal integration testing in a development, integration facilities, or the software systems integration facility (S/SIF). The software implementation method for testing provides assurance that engineering builds function as expected to enable smooth execution for verification and test activities. An incremental software and test approach adds the functions incrementally in a series of engineering builds. The software design/development matures through a series of engineering builds. As software is tested and demonstrated, build plans are modified for subsequent builds based on lessons learned from previous engineering builds, troubleshooting, and checkout.

The software implementation method provides assurance that software engineering builds function as expected in target software and systems environments and enables smooth execution for verification and validation activities. Disciplined software implementation principles, planning, and resources for systems buildup provide effective testing to be conducted in a development facility for a software/system integration environment. Software released under configuration management control is described in a defined documented configuration management plan (CMP) to provide the necessary requirements for software implementation inside integration facilities.

SOFTWARE INTEGRATION

All software delivered or implemented by software integration or testing is processed through a configuration and controlled software library system that maintains the official status accounting for each delivery. The integration tasks require that software design/development and test processes be in place to ensure integration is ready for team troubleshooting and checkout.

The methods for software integration provide required steps to be conducted for integration and checkout of informal software engineering builds. The software design/development team and test engineers need to develop a strategy for planning, design, execution, data collection, and test evaluation. The software integration activities are informal and flexible for software checkout to prepare for the software and systems integration phase of the work product.

SOFTWARE AND SYSTEMS INTEGRATION

The software and systems integration method provides a consistent approach to effective integration activities. The software units, components, and subsystems are assembled in accordance with defined and documented integration procedures to ensure that the software and systems elements are assembled properly. The integration levels and the development plan (DP) for software determine if constructed elements are ready and subject to verification or validation activities.

The effective methods and processes for software and systems integration require disciplined software design/development practices and test planning, test execution, configuration control, quality management, and reporting of work product testing inside integration facilities to management and the customer. Software technology books, magazines, and articles all over the world are intended to reflect “best practices” from various integration facilities supporting software companies, the military, and aerospace programs and projects. It is the responsibility of management to select effective and responsible test conductors and teams and have in place software and systems integration processes due to the importance and nature of assigned tests to be successful and provide results. 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 until issues and concerns are closed out
- Evaluating current integration facility schedules

Blocking issues can include open or electronic paperwork; unavailable software test tools; undefined systems under test; and unavailable software and test personnel. The critical roadblock is not having a plan to go forward.

SOFTWARE SUBCONTRACTOR

The software subcontractor roles and responsibilities describe how a program and projects can benefit and rely on outside companies’ resources to provide required software and hardware products to be under contract and effective. The subcontractor presentation to the customer must be understood from the start of the presentation to the end. Questions can be asked by the customer to ensure that answers meet the needs for reliability and support.

This chapter describes the methods that are performed by a software subcontractor to provide the necessary support and employment that would benefit military and aerospace programs and projects. The software subcontractor can be hired for program and project planning, configuration management, quality issues, software design/development, testing, and execution of activities or tasks related to the delivery of software work products to customers. The activities performed are in accordance with a purchase contract, and the software work products are delivered to satisfy and comply with specified acceptance and delivery requirements.

SOFTWARE AND SYSTEMS INTEGRATION DELIVERY

When it is time for software and systems integration delivery, the delivery requires integration testing to be performed to provide assurance that both software and systems are integrated and working together. The integration practices ensure that units tested are complete and documented prior to the official delivery for the customer. It is important to make the right decisions before delivery of software and system end items or hardware to the customer. At times, schedules become the priority before quality, and the lack of confidence in the customer will have an impact on future working agreements and contracts.

Make sure that systems design, program and project planning, software requirements, software design, and software and systems integration are successful and that every step or milestone has quality built in during the software design/development life cycle. Knowing problems still exist, senior, program, and project managers do not show a tick mark to show schedule accomplishments to customers. Be honest and up front with the customer and know that quality comes first; then, schedules provide the road map for the teams to produce effective work products.

The effective methods for software and systems integration will provide assurance that customer requirements are met before any thoughts about hurrying delivery. Stay the course and do not deviate from the plan. Before delivery of software and systems to customers, the following are important:

- Software media and data verification and validation are complete.
- Software documentation is released and ready for delivery.
- Necessary FAIs (first-article inspections), FCAs (functional configuration audits), and PCAs (physical configuration audits) are conducted, and all action items are closed.

PRODUCT EVALUATION

An effective product evaluation method provides the necessary process steps to conduct and perform continuous evaluations of software work products during the design/development life cycle and integration activities. Numerous product evaluation tools and checklists are developed with associated scheduled processes to perform required audits and evaluations.

The product evaluation is an integral part of program- and project-level activities that is scheduled and performed by quality software personnel on an ongoing basis. These evaluations form the basis for certification that software design/development activities have been performed in accordance with program and project plans and procedures and are in line with required quality requirements.

◀ Preliminary Activity for Week 17

Jump to...

Analysis, Application, and Exploration for Week 17 ▶

Navigation

- Home
- Dashboard
    - Site pages
    - My courses
        - Capstone Project 1
        - Network Attacks: Detection, Analysis & Counter...
        - Ojt/Practicum 1
        - Social And Professional Issues
        - System Integration And Architecture 2
            - Participants
            - General
            - 12 - Midterm Examination
            - 16 Product Evaluation
            - 17 Systems And Software Development Process Summary
            - Preliminary Activity for Week 17
            - **Lesson Proper for Week 17**
            - Analysis, Application, and Exploration for Week 17
            - Generalization for Week 17
            - Evaluation for Week 17
            - Assignment for Week 17
    - Courses

Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

Activities

- Assignments
- Forums
- Quizzes
- Resources

Bestlink College of the Philippines
College Department

Powered byeLearning Commons