Lesson Proper for Week 1

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 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. The right disciplines are identified in Figure 1.1.

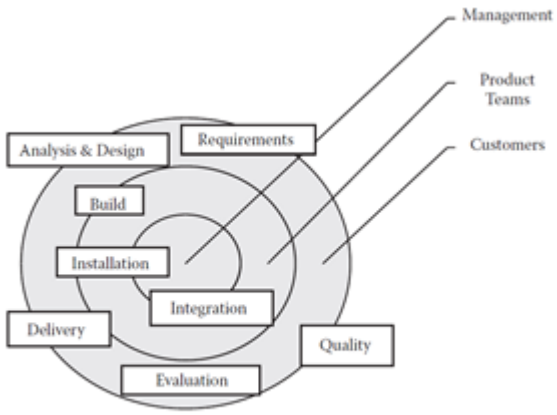

FIGURE 1.1
Start with the right disciplines.

Effective: In the software industry/companies, military and aerospace programs and projects do become effective by the implementation of achievable schedules, sound processes, and working solutions for software and systems integration.

Methods: Provide effective methods to ensure processes and tools improve productivity and prepare for the challenges that have an impact on integration environments.

Software: Software design, code, and unit tests, plans, and test procedures integrated with applied systems tell us that the software developed is done right. “Peer reviews” are key.

Systems: Accomplish allocation of software design and engineering practices for systems to be defined and documented as ready for the combination of software and systems integration.

Integration: This is the compass to combine software, systems, firmware, and hardware to work together as one.

TABLE 1.1

Planning and Engineering Task

Software Engineering Tasks	Communication	Planning	Risk Management	Deployment
Systems design	x			
Requirements		x		
Design		x		
Configuration control			x	
Integration			x	
Delivery				x
Subcontractor				x
Quality product evaluations				x

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 presented in Table 1.1 explains the disciplines and methods pertaining to communication, planning, risk management, and deployment.

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.

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.

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 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.

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.

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.

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.

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.

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.

Conclusion

Defined software disciplines include an approach or method during the software life cycle for a program and projects to provide a plan from start-up to final delivery to the customer. Many methods were discussed but the number one method is "quality first"; the other methods come in second and so on as illustrated in Figure 1.2.

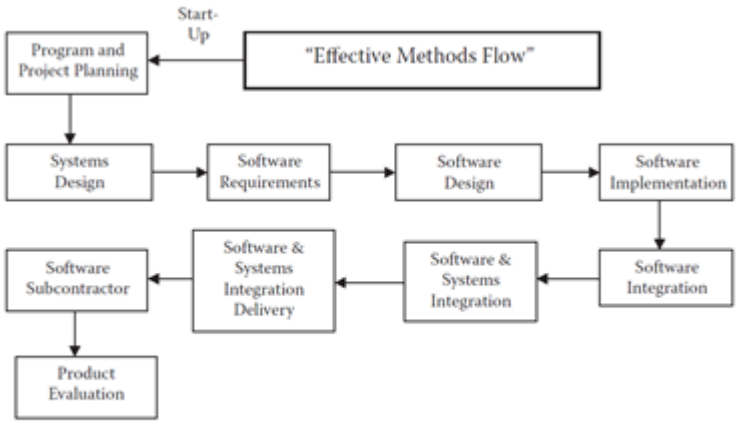

FIGURE 1.2
Effective methods flow.

Many program and project meetings are called by senior managers; in attendance are software and hardware engineers. In those types of meetings, the hardware teams will sit on one side of the table, and the software team will sit on the

opposite side. That situation is unique, but that is how it is at times. The senior managers and program and project managers attend meetings and discuss with the two teams that the software or system is not working correctly when it is time for delivery to integration facilities. There is finger-pointing, and both teams may blame the opposite

team for the problem.

The senior manager then points to the program and project managers and says, “Fix this problem.” That is why effective methods for software andsystems integration need to have hardware and software designers working

together to solve issues that could have an impact on integration, quality, and delivery schedules to the customer.

◀ Preliminary Activity for Week 1 Jump to...

Analysis, Application, and Exploration For Week 1 ▶

Navigation

Home
Dashboard
Site pages
My courses
Capstone Project 1
Multimedia
Ojt/Practicum 1
Social And Professional Issues
System Integration And Architecture 2
Participants
General
01 Introduction To Software And Systems Integration
Preliminary Activity for Week 1
Lesson Proper for Week 1
Analysis, Application, and Exploration For Week 1
Generalization for Week 1
Evaluation for Week 1
Assignment for Week 1
02 Program And Project Planning
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