Lesson Proper for Week 9

SOFTWARE INTEGRATION STRATEGY

The strategy for software integration provides a road map that describes the steps to be conducted as part of the implementation of software to start integration activities. When a strategy is planned, resources are required. This strategy should be flexible and promote an approach that could show change. Planning by senior, program, and project managers needs to track program and project progress and will require the following characteristics:

- Effective technical reviews should be conducted
- Different integration techniques and software approaches are shown
- Software designers are required to be involved from the start to the Finish

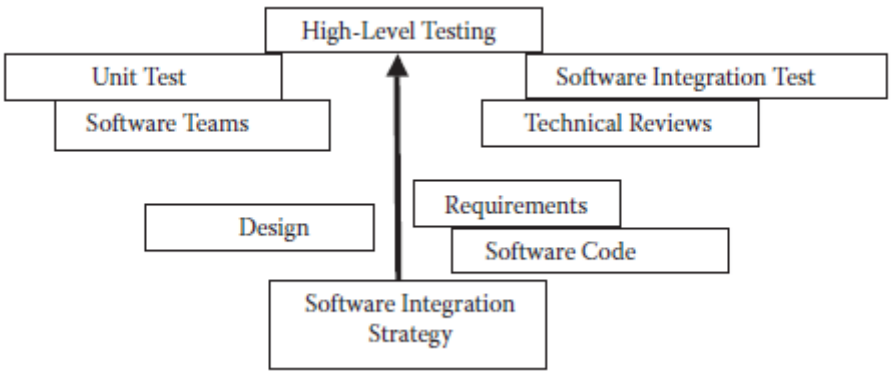

FIGURE 7.1: Software integration strategy.

The software integration strategy provides an example of higher-level integrations (Figure 7.1).

Approach to Software Integration

The approach to software integration activities is planned in advance and is the start for effective software integration. This approach accommodates lower-level integration to verify software code development that has been implemented correctly and validate major system functional expectations by customers.

The approach of effective planning for software integration provides guidance software design/development and test teams to reach milestone expectations of senior, program, and project managers. The steps for effective software integration occur numerous times as deadlines occur, and measurement problems are resolved early in schedules.

Software Integration Testing

What is software integration testing? The concept for testing software is to uncover errors, troubleshoot, and fix problems that occur during a test. Test plans and procedures are developed to test systems and, if required, rerun integration tests that are to be witnessed by quality assessors or customers.

The software test plans or procedures developed by program and project managers along with testing experts ensure that testing strategies are not wasted time during integration. Errors can appear that were previously undetected. That is the purpose of having plans and procedures in place. Test specifications are also defined and documented to provide testing steps that test conductors or that experts can implement.

Performing a review of test specifications prior to software integration testing is a strong attribute assessment before tests are complete. An effective approach to utilize a test plan or procedure for software leads to the order and discovery of errors at each stage in the test integration process.

The techniques for developing and construction of the software architecture goals take unit-tested components and build program structures established by design. The “bam theory” approach is to attempt nonscheduled software integration and testing. This approach is performed in the following three steps:

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

The Big Picture

Software processes are viewed as a spiral concept (Figure 7.2) for software integration to ensure testing is implemented for software design/ development.

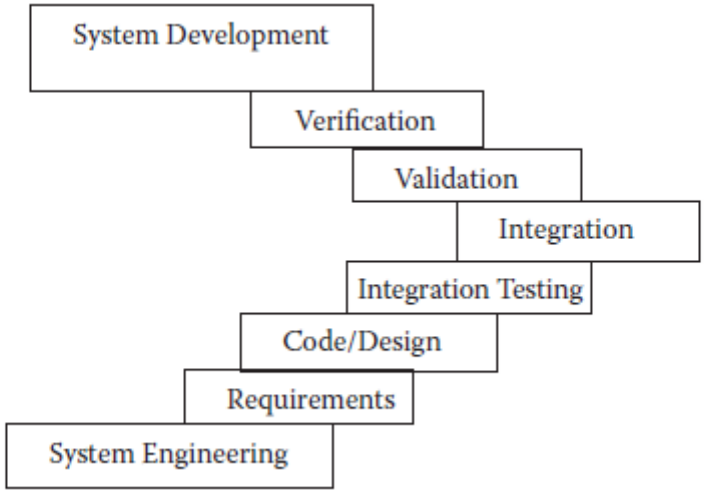

FIGURE 7.2: Spiral concept.

DEVELOPMENT FACILITY

Early in the software design/development phases for military and aerospace programs and projects, a Development Facility (DF) is normally established for software integration activities. This facility is used for the preparation of software prior to delivery to a software systems integration facility (S/SIF). Many statements or comments are made about these facilities and whether they have an effective way to test traffic loads on specific work products. In discussions with technicians and test teams, I have tried to have in place effective methods for software systems integration testing in order to show that we need traffic load tests in these development facilities.

An overview of developer facilities includes geographic locations where software integration is performed, facilities used, and secure areas along with other features. Customer-furnished equipment, software, services, documentation, data, and facilities are required according to contractual agreements along with a schedule detailing when these needed items are included. Other required resources include plans for obtaining the resources, dates needed, and availability of each resource item. The engineering design/development teams are primarily located in a designated software development geographic location.

Software Operations

There is adaptation intrinsic to software operations. Examples of this include parameter-based initialization data and settings selected or entered by a software designer/developer and test teams during operations of the software and systems retained for other test integration purposes.

The requirements for a software design/development environment must be understood when a schedule calls for software development and integration activities to be performed. Software integration plans ensures that each element of an environment performs to intended functions in support of the software design/development activities. The plans also provide requirements for test environments to perform software testing, including integration, troubleshooting, and checkout to ensure that each element of the test environment performs intended functions.

Software applications or tools used for designing, building, or integration testing the work product could be deliverable. Any non-deliverable software on which the operation depends can be identified after delivery and provisions made to ensure program and project sponsors or stakeholders obtain the same software and work product.

Software tools used for integration and hardware units installed are placed under configuration control. When software upgrades or new versions become available, program and projects evaluate and recommend whether the updates should be incorporated. Upgrades are installed as soon as is reasonable for the design/development, and integration activities are agreed to by all affected organizations. The criteria for evaluating an upgrade include considering integration problems detected, problems solved, and impacts on software integration efforts.

Software Configuration

All software configuration identifications documented in accordance with the program or project software plans are effective ways to ensure configuration control. The configured baselines identify the development life cycle, namely, functional and allocated work product baselines. Unique software documentation and media define software configuration baselines.

SOFTWARE INTEGRATION SETUP

The software integration setup method involves planning with program and project managers to coordinate with the facility operations manager. Allocated resources such as computers (i.e., workstations) and hardware units are provided to the software designer/developer and test teams to conduct informal integration testing. The software engineering builds and loading into hardware units are performed by selected build engineers.

Integration Test

Inside programs and project integration facilities, system integration tests are conducted. Verification steps ensure tests provide a check of the capabilities of software and hardware units. The software integration test is to be repeated numerous times and ensures all integration test problems are resolved and performance is accomplished early in the defined system and the system is working to software requirements.

Installation Plans and Procedures

The installation plans and procedures define systems' specification requirements. The plan and procedure for the software integration tests cover the testing of requirements and verification methods conducted in the DF. Specific integration test plans and procedures consist of checkout activities to ensure system utilization. The integration testing environment provides necessary steps to be followed, data collected, and analysis solutions are

used or implemented to produce test reports at the end of testing activities. The installation test plans and procedures are to be peer reviewed and approved for release by program and project managers to prepare for the start of software integration testing.

Integration and Checkouts

Early integration and checkouts focus on software components applied to tests to uncover errors. Once the components are tested, an informal system is constructed. Tests are executed to fix software bugs and errors. The recommendation of processing "draft-only" test plans and procedures provides loading instructions, execution, and the capabilities for uncovering problems early during integration testing. The software design and test engineers need to troubleshoot as early as possible before going into a formal test environment.

SOFTWARE INTEGRATION LOG

A 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. Facility operations managers use these logs to support operational setup activities. There are no formal released plans or procedures required during this informal phase of integration. Quality personnel are not required to support integration and checkouts performed by the software design and test teams. This is an effective method for conducting informal software testing in preparation for such activities in a facility for software and systems integration. Allow software design teams the freedom to fix and debug problems and work with test teams to ensure plans and procedures will be ready for release to support formal test phases.

This software integration log is an effective method for the software design and test team to troubleshoot problems discovered during this informal test phase. Once the software is loaded into hardware units, the software does not have an impact on and take hardware out of configuration.

Software loaded in hardware units *does not* and *will not* have an impact on and take hardware out of configuration. Hardware quality and quality software teams butt heads concerning this issue. The quality team for software provides the correct answer, so please hardware quality follow its lead. No formal software plans such as step-by- step operational paperwork or tools in the manufacturing environment are required.

SOFTWARE TEST COMPLETION

The term *acceptance testing* is discussed throughout a software integration program and projects. There are always questions when this topic is mentioned or discussed. When will the software integration testing be completed? There is never an accurate answer, and that frustrates program and project managers. The burden is always on software engineering. Remember that the importance of quality is first and not second in any software program and projects. The pressure is on when integration testing keeps going on and on and not completed in time to deliver the S/SIL work products to the customer. The metrics collected or testing models make it possible to develop guidelines for many answers to the question of when software integration and testing will be completed.

Software integration is the first phase before any stage of systems integration. Understand that metrics do come into play in the early stage of software integration and testing. All program and project managers need to implement and use metrics instead of solving problems with no data to support and make key decisions.

INTEGRATION VERIFICATION AND VALIDATION

One of the important software processes for integration is the element that is often referred to as verification and validation. The verification aspect is a set of tasks that ensure correct implementation techniques are in place to verify that the right work product is being integrated correctly. The validation concept ensures that the correct work product is the right product to validate. The quality team roles are to perform:

- Technical reviews
- Configuration management audits
- Progression monitoring during software integration
- Plans, procedures, and documentation reviews
- Qualification and acceptance testing
- Witnessing of implemented plans and procedures during integration and testing

Quality during software integration throughout the life cycle shows that proper methods and tools are being utilized. The real motive for quality can be applied for very large- and small-scale systems.

CONFIGURATION REVIEWS AND AUDITS

This step ensures all elements of software configurations are developed and are in control during software integration and test activities. Conducting and performing effective reviews and audits are key before entering into formal software and systems integration.

◄ Preliminary Activity for Week 9

Jump to...

Analysis, Application, and Exploration for Week 9 ►

Navigation

Home

Dashboard

Fair Warning

Activities

Assignments