Chapter 21: System Integration and Evaluation

This chapter covers the system integration process, system requirements verification and validation (V & V), documentation, customer acceptance testing, future trends, and guideline suggestions for a successful IT system project.

System Integration Process

The system integration process integrates hardware configuration items with software configuration items, manual operations, and other systems as necessary, into the IT system (Figure 21−1). The system developers test aggregates against their requirements as they are developed. For each requirement of the system, the system developers create a set of test cases and test procedures for conducting the system integration test. The test cases include inputs, outputs, test criteria, and use cases. The system developers evaluate the integrated system as follows:

- Coverage of system requirements
- Appropriateness of test methods and standards used
- Conformance of expected results
- Feasibility of system integration testing
- Feasibility of operation and maintenance

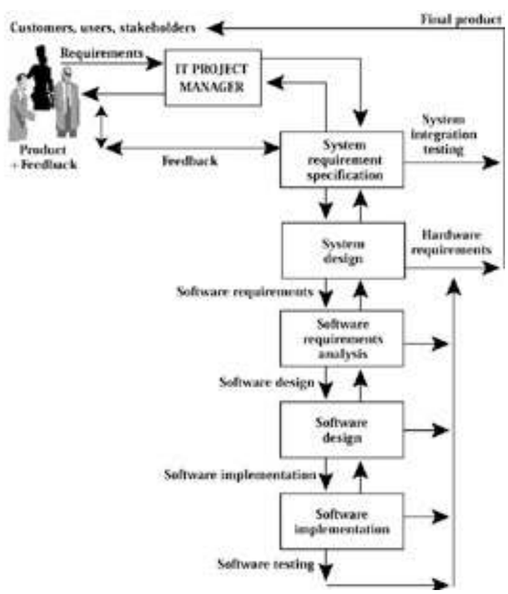

Figure 21−1: IT system integration process

Software developers install the software product in the target environment in accordance with the statement of work (SOW). The developers assist the users with the setup activities. When the installed software product is replacing an existing system, the developers support the parallel activities. The developers ensure that the software code and databases are initialized, executed, and terminated.

System Integration Testing

System developers conduct system integration testing. The IT project manager and customers witness and evaluate the results of the test. The system integration test is in accordance with the system requirements specification. It ensures that the implementation of each requirement is tested for compliance and the system is ready for delivery to the customers. The developers evaluate the IT system for the following:

- Coverage of system requirements
- Conformance to expected results
- Feasibility of operation and maintenance

The following are some of the tests and analyses:

- Input−output assertions
- Structural induction
- Regression
- Transaction flow analysis
- Stress analysis
- Failure analysis
- Concurrency analysis
- Performance analysis

The input−output assertion technique uses the assertions that are associated with the entry point, exit point, and intermediate points in the source code. The composition rule of logic lets the conjunction of assertions form along particular execution paths. This confirms that if all intermediate assertions along a given execution path are true, then the truth of the input assertion implies the truth of the output assertion for the given path.

Structural induction is based on the principle of mathematic induction. This principle is illustrated as follows: Suppose there is a set S. Assume that P is true for the first element in the set S. If P is less than or equal to N for the first element in set S, then P is true for the N 1 first element in set S.

Structural induction generally applies to recursive data structure. Regression is the retesting of the system to detect for errors introduced during modification. The system modification mandates that regression testing be performed. The developers must test the older capabilities of the system on which the customer is dependent rather than the new capabilities provided by the modification. The tests are supplemented with the other specified tests for the modifications; step−by−step testing of each affected component with the modification is of prime importance. This testing follows the logical flow of the system architecture. It involves retesting, starting from the lowest−level components, and combining these into the computer software configuration item (CSCI) and hardware configuration item (HWCI), which are ultimately threaded into the system.

The transition flow analysis identifies the transactions by drawing data flow after integration testing to model the logical flow of the system. The developers compare and confirm this analysis with the system architecture.

Stress analysis involves the behavior of the system when its resources are saturated to assess whether the system will continue to satisfy its specifications. The developers must identify the resources that can be stressed depending on a particular system. The following are some of the resources for stress test cases:

- File space
- Memory
- Input−output buffers
- Processing time
- Runtime
- Interrupt handler

Stress analysis forces a view of the system under unforeseen conditions that the user may not anticipate.

Failure analysis is an examination of the system products reaction to various types of potential failures from hardware, software, or communication. The developers examine the products specifications to determine precisely which types of failures must be analyzed and what the products reaction must be. This process assists the developers in detecting the system products ability to recover from errors, such as lost files, lost data, and duplicate transactions.

Concurrency analysis examines the interaction of the tasks that are being simultaneously executed within the system to ensure the satisfaction of the system specifications. The developers may execute concurrent tasks in parallel or interweave their execution. The developers perform this analysis during the design phase to identify such issues as potential contention for resources, deadlock, and priorities.

A concurrency analysis for implementations takes place during system testing. The developers should design, execute, and analyze tests to exploit the parallelism in the system and ensure the satisfaction of the specifications.

Performance analysis ensures that the product satisfies its specified performance objectives. The developers apply a performance analysis during each of the IT system products V & V activities. The developers perform a performance analysis during each level of testing.

The IT project manager and customers verify the test data that are constructed to correspond to the scenarios for which the performance requirements are specified. They collect, analyze, and interpret test data for the correctness of the result against the system requirements.

Statistical methods and automated test tools are available to assist for collection and analysis of the data. Data management evaluates a system for quality assurance (QA) and effective control of data. The data management process includes the following:

- Integrated data requirement list
- Instrumentation and collection methods
- Data transmission, reduction, and storage
- Analysis and reporting
- Audit trail tracing of all data decisions

The IT project manager and customer make decisions concerning the number and duration of the tests depending on resources, budget, and schedule.

Case Study 21−1: Doomed Mars Spacecraft

Missing computer codes, inadequate software testing, and a communications systems that lacked integration helped doom the Mars Polar Lander (MPL).

After losing all four Mars−bound spacecraft in 1999, officials of the National Aeronautics and Space Administration (NASA) have had to rethink their approach to using robotic spacecraft. The reports by the Mars Program Independent Assessment Team and NASAs Jet Propulsion Laboratory (JPL) Review Board will guide future mission efforts.

The JPL study, Reports of the Loss of the Mars Polar Lander (MPL) and Deep Space 2 Missions, concluded that the MPL probably crashed because of spurious signals that the MPL generated during its descent, a problem that would have been fixed by one line of missing code.The lack of communications (telemetry) to provide entry, descent and landing data for [the Mars Polar Lander] was a major mistake, the report says. Absence of this information prevented an analysis of the performance of MPL and eliminated any ability to reflect knowledge gained from MPL in future missions. The reports also confirmed that the Mars Climate Orbiter failed in 1999 because navigation data were recorded in English measuring units rather than metric.

Thomas Young, a space industry executive and former NASA official, led the assessment team. The study faulted NASA for inadequate oversight, testing, and independent analysis of the orbiter project (*Government Computer News*, 2000).

System Requirements Verification and Validation

System requirements V & V is a process of ensuring that the IT system is developed correctly. Customers, users, and stakeholders independently conduct the V & V process. Verification determines whether a system development product fulfills the customers requirements. Validation evaluates and ensures that the IT system is in compliance with its requirements. The V & V process includes formal techniques to prove the correctness of the system products. The goal of the V & V process is to ensure that the system product is free of failures and meets the customers expectations. The objective of the V & V process is to determine whether the system products satisfy their requirements and specifications in areas such as the following:

- System requirements analysis
- System design
- Quality
- Safety
- Security
- Man−machine interface
- Serviceability

The V & V process confirms whether the system is correct and error free, the product is consistent with itself and other interfacing products, everything in the product is necessary, the product is complete, and it satisfies its performance specifications. The following are some of the techniques used for V & V:

- System technical reviews
- System testing
- Proof of correctness
- Simulation and prototyping
- Requirements tracing

During the requirements and specification V & V activities, developers analyze performance objectives to ensure the following:

- Completeness
- Feasibility
    - ♦ Prototyping
    - ♦ Simulation
    - ♦ Modeling
- Testability

The developers evaluate the performance requirement during the design phase to determine whether the customers requirement is satisfied. Prototyping, simulation, and modeling are applicable techniques.

The V & V process confirms the satisfaction of the customers criteria as specified in the SOW and verifies the completion of all audits, such as functional and physical configuration. The V & V process also confirms agreement on the number and severity of unresolved system documentation errors. Once the customer is satisfied, the certification of V & V is signed and the acceptance of the system engineering, source and object code, hardware, and all related documents is complete.

Management of IT Project Documentation

Management of IT project documentation depends on the agreement between the customers and the management. The guidance and formats have already been established in the SOW. The approved and released technical documentation meets the following requirements as stated in the SOW:

- Be complete
- Establish the product baseline
- Be suitable for use in the maintenance
- Be satisfactory for accepting items produced
- Be appropriate for operational needs
- Be appropriate for logistic support purposes

Customer Acceptance Testing

Customer acceptance testing involves the proper planning and execution of tests that demonstrate that the implemented system engineering for the IT system satisfies the customers requirements. The test includes functional, performance, and stress tests that were developed during the systems development testing and integration testing. For his or her own satisfaction, the customer performs the acceptance tests to ensure that the result achieved is correct. The following are some of the tools used for acceptance tests:

- Coverage analyzer
- Timing analyzer
- Standard checker
- Statistical methods

Customers, users, and stakeholders conduct independent configuration audits, which verify compliance with requirements. The audit function validates the accomplishment of development requirements and achievements of a production configuration through the commercial items (CIs) technical documentation. The following are two types of configuration audits:

- Functional configuration audit (FCA)
- Physical configuration audit (PCA)

The IT project manager conducts these audits in conjunction with other audits such as reviews, demonstration tests, inspections, and acceptance tests. Sometimes the customer, in accordance with SOW, requires these audits.

An FCA is a means of validating the development of a CI that has been completed satisfactorily. This is a correlated prerequisite to a PCA. The manager conducts an FCA on CIs to ensure that the CIs technical documentation accurately reflects the CIs functional characteristics and necessary physical characteristics, and that test and analysis data verify that the CI has achieved the performance specified in its functional or allocated configuration identification.

A PCA is a means of establishing the product baseline as reflected in the product configuration identification and is used for the production and acceptance of the units of a CI. This ensures that the as−built configuration of a unit of a CI is selected jointly by the system developer and matches that same units product configuration identification or that the differences are reconciled. It also ensures that the acceptance testing requirements prescribed by the documentation are adequate for acceptance of production units of a CI according to the QA standards. Formal approval by the IT project manager of the CI specifications and the satisfactory completion of a PCA results in the establishment of the product baseline for the configuration item.

Management of the IT Project Audit

The IT project manager starts an audit team to work immediately after the system products are validated and verified by the customers. This team is called an *external audit team*. The team generally consists of members who were not involved in the systems development. The auditor evaluates the projects success or failure and measures the organizations achievements according to the following objectives:

- Managed the system development well
- Completed the project within the schedule and budget
- Performed the project well
- Managed the project well
- Hired a competent, effective technical team
- Satisfied customers, users, and stakeholders
- Used Internet and reuse technologies effectively
- Employed modern automated tools
- Learned lessons for future IT projects

Well−Managed, Successful IT Project

Figure 21−2 illustrates the characteristics of a well−managed, successful IT project. The cost and the risk involved influence the progress of a systems development. Before the cost increases, the chance of achieving the goals of the systems development successfully decreases. The following are characteristics of a well−managed, successful IT project:

- The customers, users, and stakeholders are satisfied.
- Fewer bugs exist in the system.
- Team members are satisfied.
- The technical aspect is handled well.
- The schedule is met.
- The IT project is completed successfully within budget and time.
- The project is well documented.

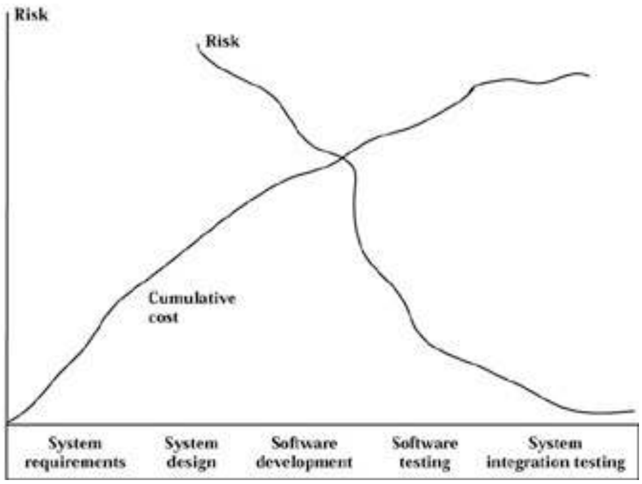

Figure 21−2: A well−managed IT project

The following is an auditors checklist for a well−managed, successful IT project:

1. Management
    a. Customers, users, and stakeholders are satisfied.
    b. The system development staff is satisfied.
    c. The system is well designed and meets the customers requirements.
    d. The system is completed on time.
    e. The system is completed within budget.
    f. The systems development is well documented.
    g. The system has a minimum level of efforts for maintenance.
    h. The IT project manager used good project management techniques.
    i. The system sets an example for future projects.
2. Technical
    a. The developers used automated computer−aided software engineering (CASE) and commercial−off−the−shelf (COTS) tools.
    b. The developers applied effective reuse.
    c. The developers used maximum Internet technology.
    d. The developers followed the team concept.
    e. The developers followed modern techniques for system development.
    f. The developers had an ongoing open communication with the project team.
    g. Supporting team assistance was readily available.
3. Operation
    a. Production execution is easy.
    b. The system has a runtime library.
    c. The system operates according to a reasonable schedule.
    d. The system has good execution documentation.
    e. The system allows for smooth conversion.
    f. The system requires a minimum man−machine interface.
    g. The system takes the users safety and security into consideration.
    h. The system is reliable and dependable.
4. Resources

a. The system is accessible and available to the customer.
b. The system has allocated time and budget for planning and control.
c. The system has a realistic schedule.
d. The system has adequate resources.
e. The system has the customers approval.
f. The system development team members demonstrate personal growth.
g. The development team members have a high morale.
h. The system has project management support.
i. The system has management recognition.
j. The system has management follow−up support.

Unsuccessful IT Project

Figure 21−3 illustrates characteristics of an unsuccessful IT project. The IT project failures or nonacceptance by the customer are due to many reasons. Figure 21−3 shows a poorly managed project. The determination of the unsuccessful project requires a serious audit evaluation of the following:

- Management
- System
- Technical
- Personnel
- Resources
- Operation
- Documents
- Customers, users, and stakeholders

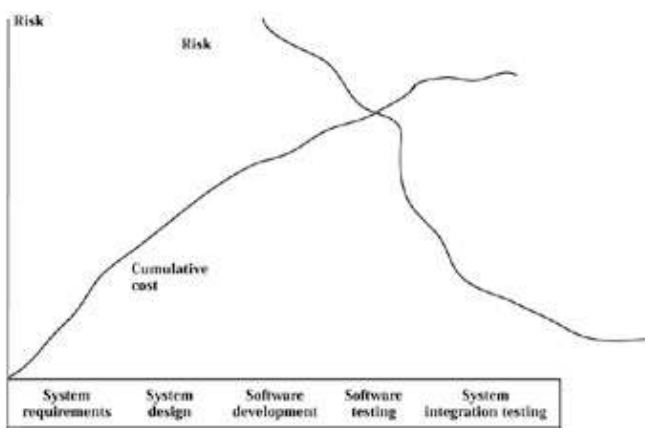

Figure 21−3: A poorly managed IT project

The IT project manager evaluates a project to determine whether lessons can be learned as a result of completing this project. The manager examines the style and techniques used during the systems development phases. The manager learns lessons by examining problems encountered in areas such as planning, manpower, budget, scheduling, estimation, and organization. He or she should apply methods that can help in avoiding the recurrence of such problems in future IT projects. The following are the main evaluation activities of the manager:

- Redetermine objective and scope.
- Review the project book.
- Review tasks and activities.
- Review assignment of responsibilities.

- Reevaluate planning, scheduling, and estimating.
- Reevaluate budgeting.
- Reevaluate risk factor management.
- Reevaluate organization.

The manager studies the system evaluation to determine whether the system was feasible, achieved its objectives, or met the customers requirements. The system evaluation also determines the following:

- Whether the system development met the design criteria with regard to solving the customers requirements
- Whether the technical aspect of the systems development met the performance objectives in terms of resources; response timing; principles; and goals of system development, safety, and security
- Whether the cost objectives were met in terms of recurring operating costs
- Whether the data flow procedural system was performed effectively for the customer
- The extent of the customers satisfaction with the systems development
- The effectiveness of the man−machine interfaces
- Whether the user guides, operator notes, and other necessary documentation are adequate to enable full and proper use of the system
- Whether the system documentation and procedures are adequate to maintain the system

The following are some of the questions that the IT project manager should ask during a technical evaluation:

- Does the technical evaluation confirm the use of proper methodologies in system and software development?
- Did the developers use modern automated tools and CI?
- Did the developers use the system modeling and simulation technique?
- Did the developers use the Internet for reuse assets?
- Did the system efficiently use the reuse technique?
- Did the developers study and analyze the customers requirements properly?
- Did the developers rush the system to meet deadlines and not debug or test it properly?

Any lesson learned can help the system developers and project manager improve future IT system development.

The manager studies the personnel evaluation to learn the following:

- Was morale high?
- Were personnel properly educated and trained to cope with the stress generated by the demands of the project?
- Were personnel properly equipped to meet the challenges?
- Were resources and support groups available?
- Did the management create tension but cooperate through open communication?

The manager uses a resource evaluation to determine the following:

- Were adequate facilities provided to complete the project?
- Did the support groups cooperate in providing the necessary functions in time?
- Were lessons learned to help in future projects?

The auditor needs to reevaluate operational manuals for comparison with other successful systems. The areas of investigation are computer operations, data entry, and interfaces with the users. During the operational evaluation the auditor interviews the users and gets their feedback. The auditor investigates topics of man−machine interfaces, timing, and manpower requirements in data entry and operations. The auditor examines the hardware to find out if it has the capability to cope with the present volume of data and whether the quality of the result is satisfactory. The auditor reviews the systems performance, controls, equipment use, and reliability. The operation logs and the trouble reports show problems such as restarts, restoring databases, reruns due to errors, and error and switch−handling capabilities.

A document evaluation is an audit review of all of the available documentation. This includes documentation of the systems development, reviews, manuals, operational information, and correspondence between the customer and developers. This familiarizes the auditors with the project and enables them to pinpoint specific areas for investigation.

The customers evaluation is of prime importance. The auditor interviews the customer to determine the following:

- Did the customer perceive the system as successful?
- Did the customer alert the developers at any time that the requirements were not understood?
- Was the customer involved in all phases of the systems development?
- Did the customer participate in all reviews?
- Did the customer show dissatisfaction?
- Did the developers take necessary actions and efforts to satisfy the customer?
- Did the customer approve the action taken by the developer?
- Did the customers and developers have effective communications?
- What lessons can be learned to avoid problems that may occur in the future?

The auditors write reports after completion of the findings and submit them to the project manager with their recommendations. The report should be precise and to the point with supporting facts and figures. The following is an outline of the evaluation report:

1. Introduction
    a. Purpose and objectives
    b. Scope
2. Management summary
    a. Major findings
    b. Recommendations
3. Measurement of project management performance
    a. Budget (planned versus actual cost)
    b. Schedule (planned versus actual completion dates)
4. Evaluation section
5. Recommendations
6. Conclusion
7. Appendixes and exhibits

Future Trend

The future trend in the computer industry is to follow the successful footsteps of other engineering faculties. The phrases Reuse, reuse, reuse and Model, simulate, fix, test, iterate are followed to save time and cost as has been achieved in other branches of engineering. Understanding the use of modeling and simulation across the total life cycle of the IT system with the process of requirements generation, project management, design and system development, testing and evaluation, and training is key to reduction of schedule and cost. It also enables testing of the performance of the system in numerous scenarios to provide confidence to the management. The project manager learns lessons from the computer hardware industry that through practice of reuse and modeling he or she can reduce pricing by reusing architecture and components. For example, when building a house or factory, the manager does not invent new products but reuses what is available in the market to save time and cost.

A lot of information is available on the Internet. The manager should try to extract as much knowledge as possible via the Internet and reuse it. More research is being conducted to make the Internet efficient in handling visual information.

A need also exists to capture a persons experience when he or she retires. This valuable knowledge is wasted when a person retires after gaining valuable years of experience, taking the knowledgeable data with him or her. Electronic chips, knowledge−based system software, and continuous feedback for improvement do not need to be reinvented. A software mechanism is necessary to capture that valuable knowledge to be reused by others who will perform the same types of duties in the future. Instead of starting from the beginning, a new person can gain the experienced knowledge and avoid going through the same training. This process can save cost and time, make a new person on the job more experienced, and share valuable contributions in a short time. This knowledgeable data can be populated on the personal domain specific repository to be reused in the future. This system can be reused at various levels and among various different fields of life and industry.

A common saying in the industry is that a computer is as good as its design by reuse of well−tested hardware components. Computer software is also as good as its design by reuse of well−tested software components as provided by the domain engineer. They all work to save cost and time and generate a quality product.

On the Internet, wireless technology is everywhere and everything. The World Wide Web is definitely going wireless. Many companies are showing off technologies that allow the Internet to be brought anywhere and accessed through pagers, handheld devices, and cellular phones. People will be trading over the Internet, getting information, and managing their bills. New services allow consumers to receive and pay bills using a phone enabled by wireless application protocol (WAP) or a Palm VII handheld device. Desktop services also allow consumers to view summary reports and update data to money management programs such as Quicken. The system works with electronic bills and traditional paper bills, which the company scans into a computer. One reason that wireless is so exciting is that it increases the ability to be in touch with customers.

Checklist: Suggestions for a Successful IT System Project

The checklist describes why the IT system project is a victim and how to turn around a failing project. Experience shows that most of the system projects become victims of failure for the following reasons:

- The system development team did not understand the requirements of the customers, stakeholders, and users.
- The scope of the project is not well defined.
- The chosen software and hardware technology changes.
- The project lacks senior management support and commitment.
- The return on investment (ROI) is not realistic.
- The schedule for completing the project is not real.
- The users are resistant because of proprietorship of assets.

- The project lacks professionals with appropriate skills.
- The project lacks cooperation and enthusiasm among the practitioners.
- Fear exists that the success of the project is going to hurt the future business.
- Industry best practices and lessons learned are ignored.
- The project has frequent changes in manpower and management.
- Self−egotism and self−pride play a major part in hurting the project.
- The project was incorrectly presented to senior management.
- Management has a prejudice to dogmatic views.
- Customers, stakeholders, and users are not involved in the progress of the project, creating a vacuum and resistance.
- Professionals are not paid for completing the project; rather, they are compensated for the number of hours billed.
- A nonseasonal manager is assigned to manage a highly visible project.
- The project team has communication problems.
- Miscommunication exists among the key players, management, customers, stakeholders, and users.
- Reuse benefits are not properly communicated among the practitioners.
- The budget dries out.
- Milestones are not achieved on schedule.
- The project is not advertised properly.
- The testers are not involved in the beginning of the project.
- The systems QA and CM sections are not established early enough and involved in the beginning of the project.
- The IT system is not tested well.