Home > My courses > System Integration And Architecture 2 > 13 Software And Systems Integration (Cont.) > Lesson Proper for Week 13

Lesson Proper for Week 13

PLANNING

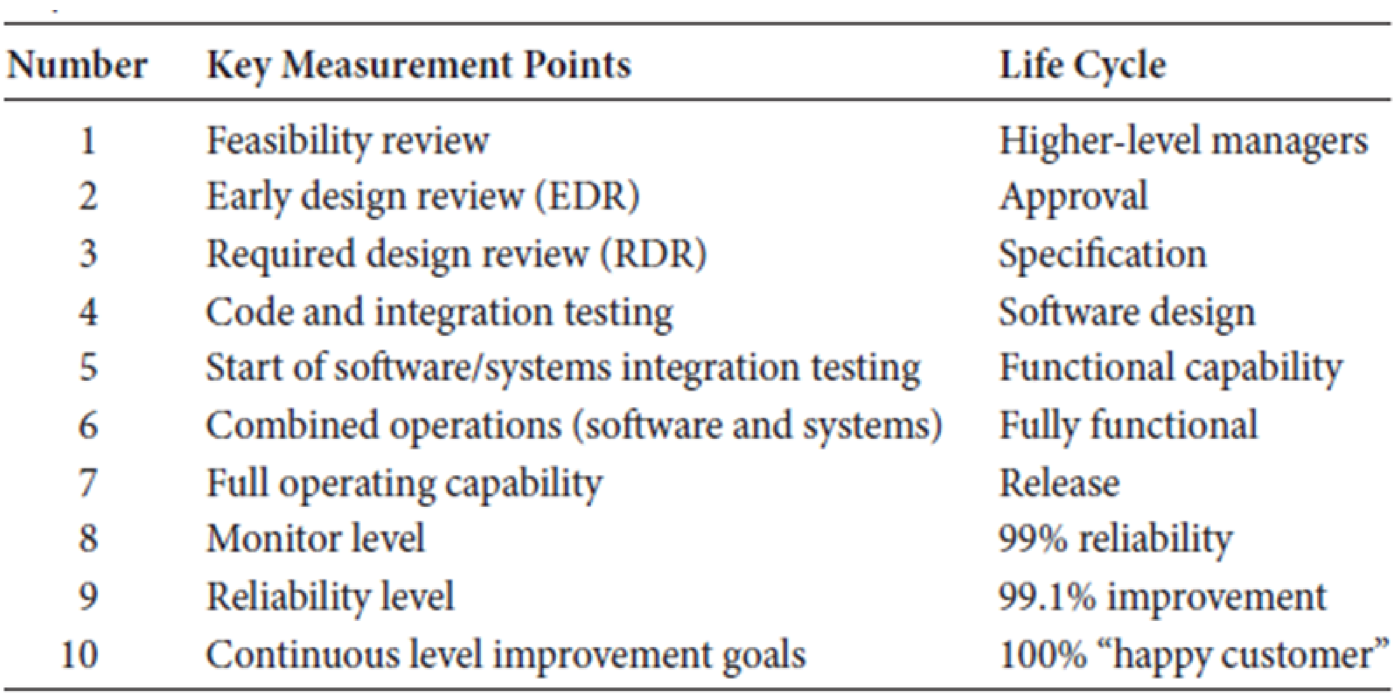
For planning, develop the SSIP and strategy to understand the systems you integrate, including the environment, functions, and constraints. Ensure requirements are testable, operational, and technically realistic. Consider using an integration readiness review plan for operational criteria in the integration environment. The planning for software and systems integration activities involves everyone from the start, including subcontractors and customers. The programs and projects require integrated processes per released software plans, installation, and checkout procedures. Before conducting software and systems integration, lab operations implement a readiness review to ensure that trained personnel are available and lab environments are ready for integration activities.

Ø Monitor Planning Progress

There have always been ways to track and monitor progress for planning on software programs and projects. The planned resources of building and supporting software and systems integration activities ensures that development time and effort (cost and staff) are in place and receive the go-ahead for implementation. The projections of not only program and project managers but also higher-level managers are important to ensure and provide broader responsibilities.

These managers are responsible for providing resources (i.e., staff, managers, funds, development time, etc.) to enable the pieces to be in place for programs and projects to follow plans and procedures. They will need to know the amount of personnel that will be available months ahead of time to evaluate the ability of the programs and projects to undertake new work.

Key measurement points are called milestones. They occur at points in the software design/development life cycle as suggested in Table 8.1. Another planning process to consider is to apply statistical control to the software design/development life cycle. Many programs and projects are familiar with this concept to manufacture work products. This tool or concept can be implemented for software and systems integration.



Number	Key Measurement Points	Life Cycle
1	Feasibility review	Higher-level managers
2	Early design review (EDR)	Approval
3	Required design review (RDR)	Specification
4	Code and integration testing	Software design
5	Start of software/systems integration testing	Functional capability
6	Combined operations (software and systems)	Fully functional
7	Full operating capability	Release
8	Monitor level	99% reliability
9	Reliability level	99.1% improvement
10	Continuous level improvement goals	100% “happy customer”

TABLE 8.1: Key Measurement Points

Measurements that fit within these control limits can reflect instability of progress. Some processes are still good, but sometimes the processes fall outside control limits.

Apply the principal of statistical control to a knowledge process, such as having a projection of what is expected to be completed with a low defect rate. The programs and projects at times follow expected plans and procedures to ensure control. If program and project managers are assigning more personnel than planned, there will be an issue of getting back to the proposed plan, or you can preplan the program and project objectives.

The higher-level manager would want to find out what the program and project managers are doing to resolve this issue and if the customer is going to be affected or resources are needed elsewhere, leading to more time to perform integration and cost concerns.

Monitoring project programs will not fall out of the sky but should be managed instantly. The first steps are to:

§ Establish baselines

§ Collect data from previous programs and projects

§ Control work products by collecting basis metrics

§ Use teamwork so everyone cooperates to ensure customer satisfaction

Ø **Comment**

The failure to provide effective planning and coordination in preparation for integration activities will ruin planning and coordination. There is intense pressure when developed schedules require tick marks to be completed and shown to customers. For planned schedules, customers get that warm and fuzzy feeling when milestones are marked off, but to be honest, many software and systems integrations are not complete. Always tell senior, program, and project managers to be up front with teams and the customer to ensure confidence that quality comes first and then schedules will follow.

COMMUNICATION

Communication encompasses channels for passing information to support interpersonal communications along with feedback and criticism (Figure 8.3). The quality of communication in a program and project is directly related to effectiveness. When the time comes for software and system integration activities, it is essential that open information is shared at both technical and interpersonal levels.

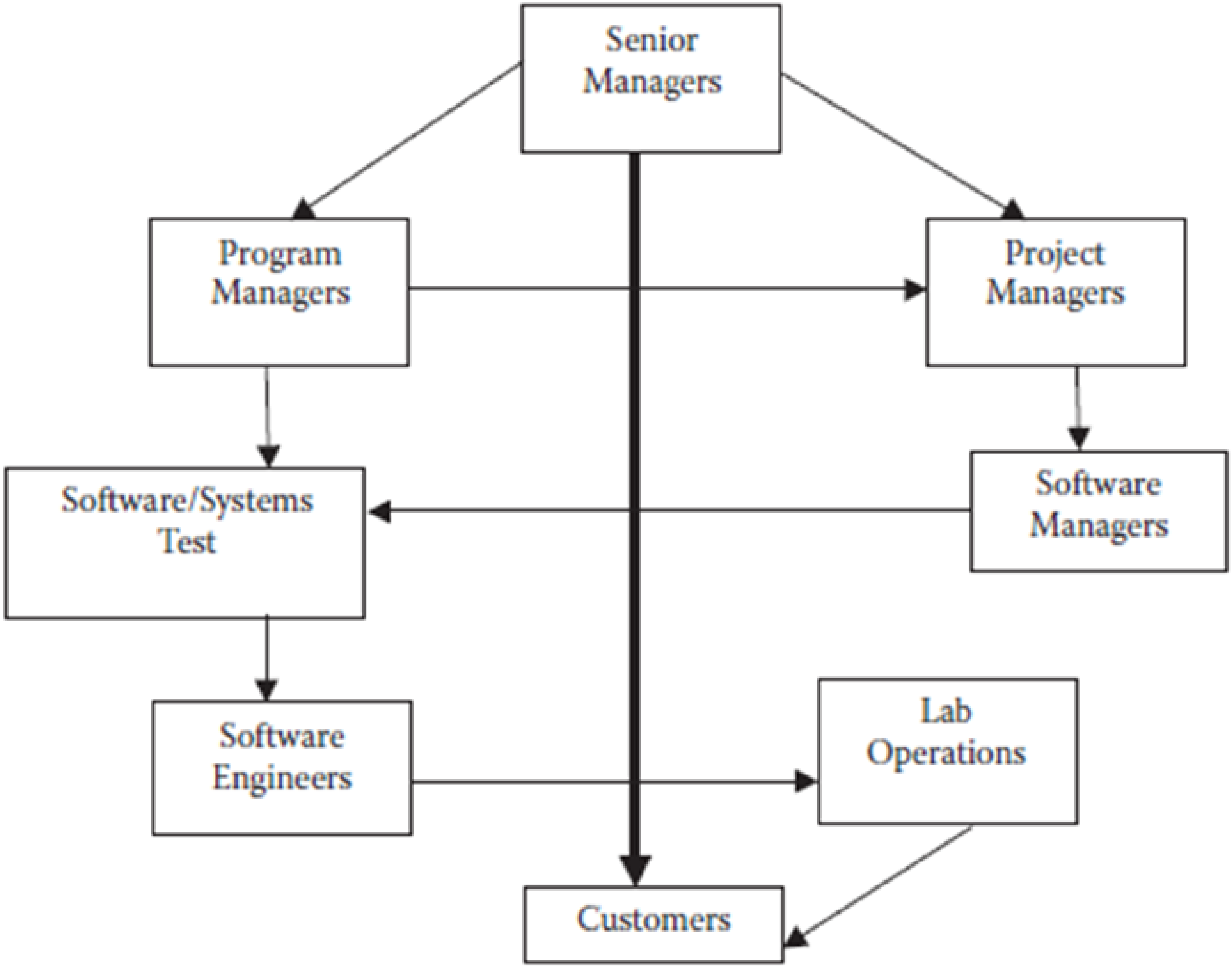


FIGURE 8.3: Communication lines.

The technical level deals with the way information describing the work product or the process is shared. On the interpersonal level, communication deals with feelings about the work product, work relationships, criticism, and personality.

Information in a program and project should be captured and communicated in writing so the understanding and coordination can be shared in the lab environment. In an electronic society in which paperwork is at a minimum, written communication can be considered formal.

RISK MANAGEMENT

It is highly recommended that risk management is conducted for integration of software and systems so it is continuous and shows the risks that occur during automation in software builds, installations, and test concerns. All risks are documented and reviewed each day during integration activities. The risk management concept is a continuous process of identification and planned team meetings to resolve and answer problems that could be found during integration. The basic process steps are summarized as follows:

- *Risk issues and concerns.* The process begins with the identification of issues and concerns. All integration teams (i.e., design, test, etc.) identify such issues and concerns through peer reviews and discussion of continuous risks so it is known what could have an impact on schedules. When possible, software subcontractor activities are included in team risk reviews. Technical performance metrics are used as the basis for risk identification and assessment.

- *Risk reviews.* Once a risk is identified, the identifying teams review the risk. A risk is rated as belonging to one of three categories: high level, midlevel, and low level. Risks rated as moderate or high require program and project risk management action presented for senior management review. Low-level risks are managed within teams and are reviewed regularly to ensure risk mitigation.

- *Risk management plans.* After a risk is defined and assigned to a team, that team will develop and implement risk management plans and continue to assess risk status until the risk is addressed and closed.

- *Risk monitoring.* For risks assigned to teams, the team provides the risk status using the risk management database. The team lead manages and maintains the database for tracking and reviews. This database generates status charts and reports for programs, projects, and customer reviews.

Ø Risk-Based Integration

Once the program and project managers agree on the estimates to create a plan for risk-based integration, the plan assigns testing based on software design/development and tests. Quality is the level of risk that could affect software and systems integration activities. Risk-based integration is reviewed when analysis is performed to root out software design/development and test defects. During integration and analysis, the test team allocates development and execution efforts based on risk. The procedures used are based on reactive techniques to detect and sort out high-risk areas. When test results are released, test cases executed, and bugs found during integration, you are able to trace the quality risks.

Ø **Risk Integration Standards**

Examples of how risk integration standards, including those for quality, apply to embedded software that controls software and systems is identified in ISO/IEC (International Organization for Standardization/International Electro technical Commission) standard 61508. This standard focuses on risks. There are two primary factors that determine the level of risk:

§ Likelihood of problems occurring

§ Impact of problems that could occur

Technical ideas such as coding and unit tests are the problems that arise when likelihood concepts come into play.

During a program and project, we must reduce risk to a tolerable level when applications are software improvements to a system or hardware unit. We have to build quality from the beginning and not at the end by making defect-preventing actions to software requirements, design/development, and integration testing. Risk integration standards require software requirements and test design to be structured. Hardware units are visible, but inside these units is software that controls the hardware so it comes alive. The movement of the hardware and software requires multiple levels of testing.

REQUIREMENTS

The teams define and develop software requirements that are selected for implementation and completion during software and systems integration. Completeness and accuracy for software requirements are verified with key work product developers. The customer should always be included in the definition of the requirements to ensure there is complete and concise understanding for their business needs.

Problems discovered in defining and developing the requirements for software are coordinated with higher-level system personnel and fixed quickly to make sure schedules are not impacted for the release of the work product. Derived requirements come into play when the performances of software are defined and applicable to systems design needs for delivery of the software work product for software and systems integration activities. The definitions of software requirements are documented in the development plan for process and work product standards. The measurement of data and metrics generated are reviewed and verified for completeness by program and project plans.

All software requirements are identified for the automation of builds and installations inside the software and systems integration environment. The software work products are integrated to be correct and reflect continuous improvement.

Ø **Evidence of Requirements**

Conformance to software requirements shows evidence that program and project-developed software and commercial off-the- shelf (COTS) or non-development items (NDIs) elements are defined and documented. The documentation of installation procedures shows the evidence utilized for the automation of software build tools. When subcontractors provide software, those elements are identified by approved plans for use during software and systems integration.

◀ Preliminary Activity for Week 13

Jump to...

Analysis, Application, and Exploration for Week 13 ▶

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
      - 13 Software And Systems Integration (Cont.)
        - Preliminary Activity for Week 13
        - **Lesson Proper for Week 13**
        - Analysis, Application, and Exploration for Week 13
        - Generalization for Week 13
        - Evaluation for Week 13
        - Assignment for Week 13
  - 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