Unit 6: System Installation

AGENDA

- Planning, Equipment Installation,
- Program Developments,
- Design and Documentation of Software,
- Program and System Testing,
- Errors, File Conversions,
- User Training,
- Performance Evaluation of the System,
- Quality Assurance,
- Post-Implementation Review.

Planning and Equipment Installation

- Installation is the process of moving from the current system to the new or enhanced system. This is also refer to conversion activity; converting an old system to a new system.
- At this stage, all users must give up their reliance on the current system and begin to rely on a new system.
- There are four approaches of installation; direct, parallel, single-location and phased installation.
- Which approaches will be used is depends on the organizations and the system's scope and complexity.
- **Types of software installation**
- In the past, most off-the-shelf software came on CDs or DVDs. Many types of software are now available to buy and download online, and are simple to install.

Planning and Equipment Installation

- You can install software on individual machines or on network servers - this is common in situations where several employees and applications share the software.
- For businesses with more than ten desktop PCs, it is often better to set them up with a common set of software. This makes it easier to update and check that software is properly licensed.
- You may need to activate certain types of software after installation, often online or by phone. If you change the hardware details of the PC, you may need to re-activate the software.
- If you don't want to install software on your PCs or servers, you can opt for **software as a service solutions** and gain access to software applications entirely over the internet.

Planning and Equipment Installation

- **Software upgrades**
- Software upgrades are generally needed to ensure that your software continues to perform properly and securely.
- Software upgrades may become available because:
  - New or improved functionality is added to the software
  - Current software has become incompatible with other, newer programs
  - Security 'fixes' or 'patches' are needed to address 'bugs' discovered in the software
- You can usually access upgrades online. It is important to install them and keep systems software fully up to date.
- Otherwise, your systems may be vulnerable to newly discovered security flaws.
- If you're using software stored in the cloud, your service provider will manage your upgrades. See advantages **of cloud computing**.

Planning and Equipment Installation

- **Software support**
- An important factor in choosing software is the availability of technical support. When assessing support, you should consider:
- **Location** - Time differences can affect availability during particular times. Some providers may offer online or email support, or real-time messaging.
- **Price** - You may get a period of free technical support when you install your software.
- However, software support is usually charged for after a certain period of time or after a certain number of queries.
- Check the cost carefully, especially if the support is on a long-distance or premium-rate phone line.

Planning and Equipment Installation

- **Response times** - If your business depends on your software, fixing problems quickly may be vital.
- Always read the small print of your maintenance contract.
- Some suppliers may stop offering support for older versions of software which can make it difficult to keep software secure.
- Look around for the best deal, but make sure that the supplier you choose can deliver on its promises.
- Find out how to choose and manage your IT supplier.

Program Development

- Coding is the process whereby the physical specifications created in the previous phases are turned into working computer codes by the programmer team.
- Coding is an activity where all the designed during the previous phases will be programmed using a software that have been defined before.
- During the coding, each program module will be tested individually, then as part of a larger program, and as the complete system.
- At the same time, the team should come out with a system and user documentation to help users in using the system.

Program Development

- There are several sub activities involved during coding activity:
  - Write and test new programs
  - Network building and testing
  - Database building and testing
  - Install and test new software

Software Design and Documentation

- Well – designed, modular software is more likely to meet the maintenance, reliability, and testing requirements.
- Three specific tools are used: Structured flowcharts, HIPO diagrams, and Warnier / Orr diagrams.
- **Structured Flowcharts**
- Structured flowcharts, also called Nassi-Schneiderman Charts, are graphic tools that force the designer to structure software that is both modular and top- down.
- They provide a structure that can be retained by programmers who develop the application software.
- There are three basic elements used in developing structured flowcharts: process, decision, and iteration.

Software Design and Documentation

- **HIPO**
- HIPO is another commonly used method for developing systems software. IBM developed this method of Hierarchical Input Process Output (HIPO), for large, complex operating systems.
- The visual table of contents (VTOC) shows the relation between each of the documents making up a HIPO package.
- There is one diagram, functional diagram, for each box in the VTOC. Each diagram shows input and output (right to left or top to bottom), major processes, movement of data, and control points.

Software Design and Documentation

- **Warinier/Orr Diagrams**
- Warnier/Orr diagrams (also known as logic construction of programs/logical construction of system) aid the design of program structures by identifying the output and processing results and then working backwards to determine the steps and combinations of input needed to produce them.

System Testing and Errors

- Each **program** will be tested in order to make sure it functions correctly. The philosophy behind testing is to find errors.
- Later, programs are tested in **groups**, followed by testing with the **entire system**.
- The first step in testing is to compile the program to detect any syntax errors. If there is an error, the programmers need to correct it until the program executes correctly.
- Then, the next step is to do a testing, includes **unit testing, integration testing** and **system testing**.

System Testing and Errors

- **Unit testing**
- It is a testing done at the individual level of program or module. Sometimes, it refers to module testing.
- The purpose of unit testing is to identify and reduce execution errors that cause the program to terminate abnormally, and logic errors that could have been missed during desk checking.
- **Integration Testing**
- After finishing the unit testing, the programmer will do the integration testing. It's a process of bringing together all of the modules that a program comprises for testing purposes.
- First, we need to test the root module and only one of the subordinates. After that, we need to test another two subordinates module from the same level.

System Testing and Errors

- The same process goes until for the next level and the whole system is tested as a unit. This is referred to system testing.
- **System Testing**
- After completing the integration testing, we must perform system testing, which involves the entire system. It's similar with an integration testing.
- The difference is, here, we integrate programs into system. After the system testing is completed, we can assume that the system is fully tested and are free any errors or bugs.
- So, now, it's ready to be installed in the organizations.

System Testing and Errors

- There are two general plans for testing software: The strategies of code testing ad specification testing.
- Systems are not designed as entire systems nor are they tested as single systems. The analyst must perform both unit and integration testing.
- There are other tests that are in special category, since they do not focus on the normal running of the system.
- Six tests are essential. **Peak load testing, storage testing, performance time testing, recovery testing, procedure testing,** and **human factor testing.**

File Conversion

- Conversion is a method of replacing/updating the old software/file with a new one.
- A conversion plan includes a description of all those activities which must occur during the implementation of a new system.
- The following are the activities that are included in the conversion method:
  - Name all files for conversion.
  - Identify the data requirements for the development of new files during conversion.
  - List new documents and procedures that are required.
  - Identify the controls to be used in each activity.
  - Identify the persons responsible for each activity.
  - Verify all schedules.

File Conversion

- The four methods of conversion are as follows:
  - Parallel conversion
  - Direct cutover
  - Phase-in method
  - Pilot approach

File Conversion

- **Parallel Conversion**
- In parallel conversion, both the old and the new system are used simultaneously. This method has two advantages.
- **First**, the continued use of the old system provides a fallback when the new system fails.
- Any new information system, no matter how exhaustively it has been tested, may fail or have problems shortly after implementation.
- **Second**, this method serves as an ultimate test for the new system.
- Parallel conversion is typically conducted for at least two accounting cycles. Both systems should produce exactly the same results or differences should be able to reconcile any.
- **The disadvantage** of this system is that keeping two systems running at the same time causes cost overruns.

File Conversion

- **Direct Cutover Conversion**
- In direct conversion, the new system is implemented and the old system is totally replaced.
- This method is cheap and fast, but is quite hazardous and converts from the old to the new system abruptly. As mentioned earlier, testing never uncovers all problems.
- Direct cutover works best when the conversion team anticipates the occurrence of problems and is ready to tackle them.

File Conversion

- **Phase-In Method**
- This technique supports a phased approach to system conversions with the ability to come up with improved and more accurate estimates of the amount of work to be undertaken, thereby avoiding large contingencies for risk management.
- The phase-in method is used when it is not possible to install a new system throughout the organization at once.
- The long-term phase-in periods **create difficulties** for analysts whether the conversions go well or not. If the system is working well, early users will communicate this information to other personnel who are waiting for implementation.
- On the other hand, if there are problems in the early stages of implementation, this will also **spread among the users**. This may lead the users to react negatively to the smallest mistakes, even their own.

File Conversion

- **Pilot Approach**
- When new systems also involve new techniques or drastic changes in organization performance, this approach is preferred.
- In the pilot approach, a working version of the information system is implemented in one part of the organization.
- When the system is approved for its performance, it is installed throughout the organization all at once or one stage at a time.
- **Types of Conversion**
- Conversion efforts depend on the type of conversion carried out. Software conversion falls into the following categories: **DBMS type, language type** and **network type.**

File Conversion

- **Conversion Strategy**
- To carry out a work where so many factors get involved, it is always better to follow a plan for conversion.
- **Hardware Conversion Strategy**
- It requires working on system hardware. All hardware may not support all kinds of software. The conversion strategy should be such that the existing system does not stop functioning or need not be brought to a halt.
- **Software Conversion Strategy**
- A section is to be devoted to software conversion strategy describing strategies to be followed in case of software.
- Hardware is not much varied as compared to software. Each hardware or peripheral comes with its driver software.
- It is also the fact that all hardware do not support all software.

File Conversion

- **Data Conversion Strategy**
- All the factors related to assurance of data quality and controls in data conversion are to be clearly mentioned.

User Training

- An essential part of the computer-based information system development is **end user training**.
- Many training methods are being made use of like, both instructor-led and self-paced. Most of them are aimed at providing the end-users with the required skills.
- User training may involve **equipment use**, particularly in the case where the individual involved is both operator and user.
- User training must also instruct individuals in **troubleshooting** the system, determining whether a problem is **caused by the equipment** or **software** or by **something they have done** in using the system.

User Training

- Including a troubleshooting guide in systems documentation will provide a useful reference long after the training period is over.
- Data – handling activities receiving the most attention in user training are
  - Adding data (how to store new transactions),
  - Editing data (how to change previously stored data),
  - Formulating inquiries (finding specific records or getting responses to questions) and
  - Deleting records of data.
- The bulk of systems use involves this set of activities, so it follows that most training time will be devoted to this area.

User Training

- There are two aspects to user training:
  - Familiarization with the processing system itself (that is, the equipment used for data entry or processing) and
  - Training in using the application (that is, the software that accepts the data, processes it, and produces the results).
- Weaknesses in either aspect of training are likely to lead to awkward situation resulting in user frustration, errors, or both.
- Good documentation, although essential, does not replace training.
- There is no substitute for hands – on – operation of the system while learning its use.
- The following are the current training methods:

Training Methods

- **Instructor-Led Training**
- Instructor-led training method involves trainers as well as trainees who are required to meet at the same time, but not necessarily at the same place.
- Two broad categories of this type of training are: virtual and normal classrooms.
- **Virtual Classroom**
  - The characteristics of a virtual classroom are:
  - The trainees and trainers must meet at the same time, but not necessarily at the same place.
  - The training session could be collaborative or one-on-one.
  - The tools used for training could include videoconferencing, text-based Internet relay chat tools or virtual reality packages

Training Methods

- The following are the merits and demerits of virtual training:
- **Merits**
- This technique helps save money, which is otherwise spent on travelling.
- The outcomes of this technique are similar to the face-to-face method but cheaper.
- **Demerits**
- The interaction between trainer and trainee is still in a limited context.
- The technologies must be sufficient to support this category of training.
- A test of the objectives for the training is hard to conduct.

Training Methods

- **Normal Classroom**
- The features of a normal classroom are as follows:
- The trainers and trainees must meet at the same time and at the same place. The training session could be either collaborative or one-on-one.
- The tools used to provide training are the same as those used in the normal classroom (i.e., blackboard, overhead projectors, Liquid Crystal Display (LCD) projector, etc.).
- **Merits**
- The user's questions should be answered by experts.
- The context of the training is unlimited. Information can also be transferred by the trainer using computer technologies.
- Most of the courses guarantee a certificate for the trainee announcing their ability following course completion.
- The abilities of the trainees are easy to test.

Training Methods

- **Demerits**
- It is an expensive technology as employers need money to meet travel expenses.
- In addition, the travel time leads to the employee's downtime being extended.
- The trainees involved in face-to-face training will have less interest and perceived learning than those who participate in a computer-based training.
- It is difficult to set up this method as it requires time and resources.

Training Methods

- **Self-Paced**
- This method of training involves both trainers and trainees, who do not require to meet at the same time or at the same place.
- This type of training can be divided into two main classes, that is, multimedia and Web-based. The details for each category are as follows:
- **Multimedia**
- This category has the following features:
- The skills are learnt by the trainees themselves, at their own pace and in their own time.
- This category of training is commonly developed and stored in a CD-ROM (Computer Disc-Real-Only Memory) where it will run on a PC (Personal Computer).
- Normally, the courses are presented in a multimedia format.

Training Methods

- The following are the merits and demerits for this category of training:
- **Merits**
- Trainees have the option of accessing the courses at their convenience.
- Certain courses also let the instructors change the outline of the course easily by allowing users to modify the graphics, the screen layout and training questions without help from external programmers.
- This could help to reduce the cost in developing an in-house training course.
- It can be used to provide JIT training to the end-users.
- The courses are normally represented in a multimedia form. Thus, they can stimulate the users' interests.

Training Methods

- **Demerits**
- The training must be imparted in a conservative manner as this technology cannot be used completely in lieu of an instructor.
- PC-based training is only appropriate for conveying the basic information.
- It is not easy to test the abilities of the trainees.

Training Methods

- **Web-Based Training**
- This category has the following features:
- It lets trainees learn at their own pace and convenience.
- It is capable of supporting both the Internet and Intranet.
- The structures of the courses are often presented in hyper-media format.
- **Merits**
- Trainees can access the courses at their own convenience.
- Online training lets the employees spend more time in the office.
- It can be used to provide JIT training for the end-users.
- Web-based training technology permits organisations to tailor training requirements to fulfil individual needs.
- This technology results in cost reduction as the users are not required to pay for travel expenses and the employee's downtime is shorter than ever before.

Training Methods

- **Demerits**
- The medium has its limitations and the courses are still in the early phase of development.
- The developer will, therefore, be forced to minimize the video context and use a text-based style for the instructions.
- On its own, this technology cannot substitute instructor-led training. Therefore, the training should be given conservatively. There is no simple manner of conducting a test of objectives of the courses.

Quality Assurance

- Quality assurance is the review of software products and related documentation for completeness, correctness, reliability, and maintainability.
- And, of course, it includes assurance that the system meets the specifications and the requirements for its intended use and performance.
- Four levels of quality assurance: testing, verification, validation, and certification.

Quality Assurance

- **Testing**
- Systems testing is an expensive but critical process that can take as much as 50 percent of the budget for program development.
- The common view of testing held by users is that it is performed to prove that there are no errors in a program.
- However, this is virtually impossible, since analysts cannot prove that software is free and clear of errors.
- The tester, who may be an analyst, programmer, or specialist trained in software testing, is actually trying to make the program fail. A successful test, then, is one that finds an error.

Quality Assurance

- **Verification and validation**
- Like testing, verification is also intended to find errors.
- **Verification** testing includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product.
- It is also known as **static testing**, where we are ensuring that "we are developing the right product or not". And it also checks that the developed application fulfilling all the requirements given by the client.
- **Validation** refers to the process of using software in a live environment on order to find errors.
- **Validation** many continue for several months. During the course of validating the system, failure may occur and the software will be changed.

Quality Assurance

- **Certification**
- Software certification is an endorsement of the correctness of the program, an issue that is rising in importance for information systems applications.
- There is an increasing dependence on the purchase or lease of commercial software rather than on its in-house development.
- For example, selected accounting firms are now certifying that a software package in fact does what the vendor claims it does and in a proper manner.
- They do not, however, certify that the software is the right package for a certain organization. That responsibility remains with the organization and its team of analysts.

Performance Evaluation of the System

- Evaluating a system includes the hardware and software as a unit. Hardware selection requires an analysis of several performance categories.

1. **System availability**. When will the system be available?
2. **Compatibility**. How compatible is the system with existing programs?
3. **Cost.** What is the lease or purchase price of the system? What about maintenance and operation costs?
4. **Performance**. What are the capacity and throughput of the system?
5. **Uptime.** What is the ‘uptime’ record of the system? What maintenance schedule is required?
6. **Support**. How competent and available is the vendor’s staff to support the system?
7. **Usability**. How easy is it to program, modify, and operate the system?

Performance Evaluation of the System

- For the software evaluation, the following factors are considered:

1. The programming language and its suitability to the application(s).
2. Ease of installation and training.
3. Extent of enhancements to be made prior to installation.

Performance Evaluation of the System

- In addition to hardware/software evaluation, the quality of the vendor's services should be examined.
- Vendor support service include the following:

1. **Backup**. Emergency computer backup available from vendor.
2. **Conversion**. Programming and installation service provided during conversion.
3. **Maintenance**. Adequacy and cost of hardware maintenance.
4. **System development**. Availability of competent analysts and programmers for system development.

Post-Implementation Review

- In-process reviews are informal reviews intended to be held during the conduct of each SDLC phase and there is little reporting to the customer on the results of the review.
- Unlike the in-process reviews which deal with a single product phase end reviews include examination of all the work products that are to be completed during that phase.
- Phase-end reviews permit the user or customer to verify that the project is proceeding as intended or to give redirection as needed.
- They are also major reporting points for software quality to indicate the management how the project is used to adhere the standards, requirements and resource budgets.

Post-Implementation Review

- Two formal audits, the **Functional Audit (FA)** and the **Physical Audit (PA)** are held, at the end of the SDLC (Software Development Life Cycle).
- They are the final analysis of the software product to be delivered against its approved requirements and its current documentation respectively.
- The FA compares the software system being delivered against the currently approved requirements for the system, usually through an audit of the test records.
- The PA is intended to assure that the full set of deliverables is an internally consistent set, i.e., the user manual is the correct one for this particular version of the software.
- The PA relies on the records for the delivered products.

Post-Implementation Review

- The PIR usually is conducted six to nine months after implementation. Its purpose is to determine whether the software has met the user's expectations for it in actual operation.
- Data from the PIR is intended for use by the software quality practitioner to help improve the software development process.
- Table follows summarizes some of the approaches used in PIR.

Approaches	3 To 6 Months after Software System Implementation
Software system goals versus experience	This approach supports return on investment, schedule results, user response and defect history.
Review results usage	This approach supports input values for process analysis.

Post-Implementation Review

- The post implementation review ensures that the project has met its objectives by evaluating the development and management processes of the project.
- It also enables you to compare actual expenditures on the project with that to the estimated expenditure of a project.
- A PIR **highlights the strong and weak points** of a project.
- Following are the objectives of a PIR:
    - To demonstrate the **success** of a project **against** the projected **costs, benefits and timelines** in the business case.
    - To identify the opportunities to add **additional value** to the project.
    - To identify **strengths and weaknesses** of the project for future reference and appropriate action.
    - To make **recommendations** on the **future** of the project.

Post-Implementation Review

- **Post Implementation Review Process**
- The process begins during the initial stages of the project development.
- The project team and the primary users of the project define the **performance measures** and **benefits expected** at the time of planning of the project.
- The approving authority then approves these performance measures and expected benefits.
- Finally, the project team **ensures that appropriate measures are implemented** so that project performance can be maintained throughout the life cycle of the project.
- Upon completion of a major project, they prepare a post implementation evaluation review.
- The project manager is responsible for implementation process.

Post-Implementation Review

- The staff members that should be included in the review process are Project team and management, user staff, strategic management team and external users.
- Elements of post implementation review are project history, background, cost history, project management methodology, performance measures, implementation experiences, and audit issues.