**COMP 2831 Week 10 Exercise -A01029917 Andrew Hewitson**

**Chapter 11**

**1. List and describe the main tasks that are performed during systems**

**implementation.**

* **Application Development –** analysts determine the overall design strategy and work with programmers to complete the design.
* **Testing –** a plan is designed by a system analyst that outlines test steps and test data for integration testing and system testing
* **Documentation –** A description of a system’s functions and how they are implemented. Prepared mostly during the design phases. Includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project.
* **Training –** A plan is designed and implemented that outlines required training for users, managers, and IT staff members. The plan must outline the requirements for these people to understand the system and know how to use it effectively.
* **Data Conversion –** Existing Data is loaded into the new system, transformed as needed. Depending on the system, data conversion can be done before, during or after the operational environment is complete.
* **System Changeover –** The process of putting the new information system online and retiring the old system. Changeover can be rapid or slow, depending on the method.
* **Post-Implementation Evaluation –** An assessment of the overall quality of the information system. The evaluation verifies that the new system meets specified requirements, complies with user objectives, and achieves the anticipated benefits. This helps improve future projects.

**5. What is cohesion? What is the difference between close coupling and loose**

**coupling?**

* **Cohesion –** A measure of a module’s scope and processing characteristics. High cohesion means a module only performs one task. The lower
* **Loose Coupling –** less dependant on other classes / modules. Easier to maintain and modify because the logic in one module does not affect other modules.
* **Close Coupling –** Highly dependant on other classes / modules. A change in one class can ‘break’ things in many other classes.

**6. Describe three main types of testing and the order in which they are performed.**

1. **Unit Testing –** The testing of an individual program or module. Goal is to eliminate execution errors that could cause the program to terminate abnormally, and logic errors, which cause the program to return faulty information, that could have been missed during desk checking.
2. **Integration Testing –** Testing two or more modules that depend on each other
3. **System Testing –** Last is system testing. A form of testing involving the entire system and includes all typical processing situations. Sources of input and output are tested. i.e. user input, queries, produce reports. All processing options are verified by users and the IT project development team to ensure that the system functions correctly.

**8. What is the difference between an operational environment and a test environment?**

* **Operational Environment -**  As close to the finished product and bug free as possible. This is the environment for the actual system operations which the customer will be operating in. It includes hardware and software configurations, system utilities, and communication resources. It contains live data and is accessible only by authorized users.
* **Test Environment –** Once the developers feel the software performs according to requirements it is passed into the test environment. Separate from the operational environment. It allows the analysts and programmers use to develop and maintain programs without affecting the live system. It exists in a separate limited access workstation or server located in the IT department.

**9. List and describe the four main system changeover methods. Which one generally**

**is the least expensive? Which is the safest? Explain your answers.**

* **Direct Cutover –** Causes the changeover from the old system to the new system to occur immediately when the new system becomes operational. Generally, **least expensive** because only one system must be maintained at one time. (Although could be the most expensive if catastrophic errors occur).
* **Parallel Operations –** Requires that both the old and the new systems operate fully for a specified period. Data is input into both systems and output from the new system is compared with the old system. This is the safest because the company can use the old system as a backup. Most costly changeover method.
* **Pilot Operation –** implementing the complete new system at a selected location of the company. The old system continues to operate for the entire organization, including the pilot site. Restricting the implementation to a pilot site reduces the risk of system failure. Less expensive than a parallel operation for the entire company.
* **Phased Operation –** Implement the new system in stages, or modules. Combines direct cutover and parallel operation to reduce risk and costs. Only a part of the system is given to all users, while pilot operation provides the entire system, but to only some users. Risk of errors or failures is limited to the implemented module only. Less expensive than full parallel operation because you must work with only one part of the system at a time.

The safest method would be a phased operation implementing the parallel operation method. It would give analysts time to discover errors incrementally as the system is rolled out and unlike the pilot operation method you could see how the system will work across the whole company. (Because what is good for Denver might not be OK everywhere.) This method would be the most labour/time intensive and the costliest but would be the safest.