PROTOTYPING, INCREMENTAL DELIVERY AND ALBRETCH FUNCTION POINT ANALYSIS

SOFTWARE PROTOTYPING

- A prototype is a working model of one or more aspects of the projected system.
- It is constructed and tested quickly and inexpensively in order to test out assumptions.
- Prototype can be classified as throw-away or evolutionary.

REASONS FOR PROTOTYPING

- Learning by doing
- Improved communication
- Improved user involvement
- Clarification of partially known requirements
- Demonstration of the consistency and completeness of a specification
- Reduced need for documentation
- Reduced maintenance costs
- Feature constraint
- Production of expected results

DRAWBACKS AND DANGERS

- Users can misunderstand the role of the prototype.
- Lack of project standards possible.
- Lack of control.
- Additional expense.
- Machine efficiency.
- Close proximity of developers.

INCREMENTAL DELIVERY

- This approach breaks the application down into small components which are then implemented and delivered in sequence.
- Each component delivered must give some benefit to the user.
- Time-boxing is often associated with an incremental approach.

ADVANTAGES OF INCREMENTAL DELIVERY

- The feedbacks from early increments improves the later stages.
- The possibility of changes in requirements is reduced.
- Early delivery of some useful components improves cash flows, because you get some return on investment early on.
- Smaller sub-projects are easier to control and manage.
- Gold-plating is less.
- The project can be temporarily abandoned if more urgent work emerges.

DISADVANTAGES OF INCREMENTAL DELIVERY

- Later increments might require modifications to earlier increments. This is known as software breakage.
- Software developers may be more productive working on one large system than on a series of smaller ones.

THE INCREMENTAL DELIVERY PLAN

- The nature and order of each increment to be delivered to the users have to be planned at the outset.
- The process is similar to strategic planning but at a more detailed level.
- The elements of the incremental plan are the system objectives, incremental plan and the open technology plan.

SYSTEM OBJECTIVES

- Objectives can be expanded into more specific functional goals and quality goals.
- Functional goals will include:-

1. Objectives it is intended to achieve
2. Jobs the system is to do
3. Computer/non-computer functions to achieve them

OPEN TECHNOLOGY PLAN

- This will require the use of:-

✓ A standard high-level language

✓ A standard operating system

✓ Small modules

✓ Variable parameters

✓ A standard database management system

INCREMENTAL PLAN

- For incremental plan, we have the following guidelines:-

✓ Steps typically should consist of 1-5% of the total projects.

✓ An increment should not exceed one month and should not, at worst, take more than 3 months.

✓ Each increment should deliver some benefit to the user.

✓ Some increments will be physically dependent on others.

✓ Value-to-cost ratios may be used to decide priorities.

ALBRECHT FUNCTION POINT ANALYSIS

- The basis of function point analysis is that computer-based information systems comprise five major components or ‘external user types’ that are of benefit to the users.

✓ External input types- input transactions which update internal computer files.

✓ External output types – transactions where data is output to the users.

✓ Logical internal file types- the standing files used by the system.

ALBRETCH FUNCTION POINT ANALYSIS CONT'D

- ✓ External interface file system – allow for the output and input that may pass to and from other computer applications.
- ✓ External inquiry types – transactions initiated by the user which provide information but do not update the internal files.