

SOFTWARE ENGINEERING

(MC – 310)

ASSIGNMENT – 1

AIMAN SIDDIQUA

2K18/MC/008

Explain why incremental development is the most effective approach for developing business software systems. Why is this model less appropriate for real-time systems engineering?

Incremental development is the most effective approach for developing business software systems because of the following reasons:

- The software system will be delivered quickly to the user.
- The cost for each iteration is very less.
- The testing efforts for each iteration will be less because only few additional functions need to be added in each iteration.
- The final product produced will be perfect without any defects or issues since we get feedback of the customer during each iteration and developers will resolve these defects/issues.
- This model allows user to modify/change the requirements according to his convenience at any phase of iteration.

This model less appropriate for real-time systems engineering due to these reasons:

- Generally real-time systems contain many hardware modules which are cannot be incremental and not easy to alter.
- Moreover, real-time systems are typically protection critical; it must be built on a good planned procedure.
- In Incremental development model process is not visible and system structure inclines to reduce as new increments are added. Money and time are expended on it to improve the software, regular change towards to corrupt its system structure.

To help counter terrorism, many countries are planning or have developed computer systems that track large numbers of their citizens and their actions. Clearly this has privacy implications. Discuss the ethics of working on the development of this type of system.

Countering terrorism by tracking large number of citizens and their actions can have negative implications. It is an invasion of privacy of the citizens. This can lead to leaking of private information of people which can cause misuse of sensitive information. Buying and selling of data is an important issue that needs to be considered while working on the development of this type of system.

Why do we need metrics in software? Discuss the areas of applications of software metrics. State the advantages and disadvantages of using LOC as a metric.

We need software metrics to is to determine the quality of the current product or process, improve that quality and predict the quality once the software development project is complete.

Areas of applications of software metrics:

- The most established area of software metrics is cost and size estimation techniques.
- The prediction of quality levels for software, often in terms of reliability, is another area where software metrics have an important role to play.
- The use of software metrics to provide quantitative checks on software design is also a well-established area.

Advantages of using LOC as a metric:

- It is widely used and universally accepted.
- It permits comparison of size and productivity metrics between diverse development groups.
- It directly relates to the end product.
- LOC are easily measured upon project completion.
- It measures software from the developers' point of view- what he actually does (write line of codes).
- Continuous improvement activities exist for estimation techniques.

Disadvantages of using LOC as a metric:

- Difficult to measure LOC in the early stages of a new product.
- Source instructions vary with coding languages, design methods and with programmer's ability.
- No industry standard for measuring LOC.

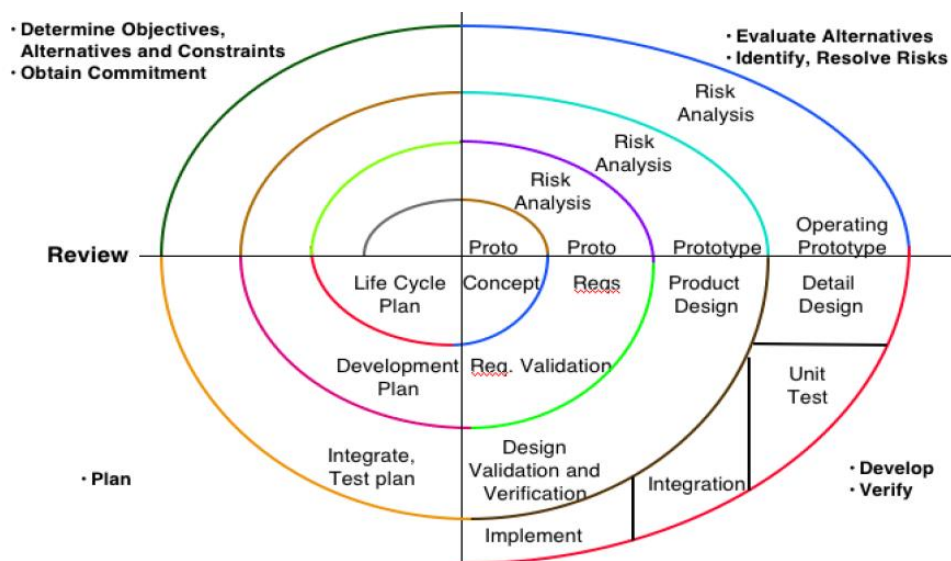
- LOC cannot be used for normalizing if platforms and languages are different.
- The only way to predict LOC for a new app to be developed is through analogy based on similar software application.

Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.

There is a fundamental difference between the user and the system requirements that mean they should be considered separately.

- The user requirements are intended to describe the system's functions and features from a user perspective and it is essential that users understand these requirements. They should be expressed in natural language and may not be expressed in great detail, to allow some implementation flexibility. The people involved in the process must be able to understand the user's environment and application domain.
- The system requirements are much more detailed than the user requirements and are intended to be a precise specification of the system that may be part of a system contract. They may also be used in situations where development is outsourced and the development team need a complete specification of what should be developed. The system requirements are developed after user requirements have been established.

Describe the spiral model of software development. Explain how both waterfall model and prototyping model can be accommodated in the spiral model.



Models do not deal with uncertainty which is inherent to software projects.

Barry Boehm recognized this and tried to incorporate the “project risk” factor into a life cycle model. The result is the spiral model, which was presented in 1986.

The radial dimension of the model represents the cumulative costs. Each path around the spiral is indicative of increased costs. The angular dimension represents the progress made in completing each cycle. Each loop of the spiral from X-axis clockwise through 360° represents one phase. One phase is split roughly into four sectors of major activities:

- **Planning:** Determination of objectives, alternatives & constraints.
- **Risk Analysis:** Analyze alternatives and attempts to identify and resolve the risks involved.
- **Development:** Product development and testing product.
- **Assessment:** Customer evaluation.

Both waterfall model and prototyping model can be accommodated in the spiral model.

- The waterfall model is accommodated where there is a low specification risk and no need for prototyping etc. for risk resolution. The activities in the 2nd the quadrant of the spiral model are skipped.
- The prototyping model is accommodated when the specification phase is limited and the prototyping (risk resolution) phase predominates. The activities in the 3rd quadrant of the spiral model are skipped or reduced in scope.