Imp

2.5 THE SPIRAL MODEL

This model is relatively a new model, proposed by Barry Boehm. It incorporates the elements of both the prototype approach along with the classic software lifecycle. The activities in this model can be organized like a spiral, that has many cycles. Typically, the inner cycles represent the early phases of requirement analysis along with the prototyping and the outer spirals represent the classic software life cycle. The spiral model is shown in figure 1.4. This model has been divided into four quadrants, each quadrant representing a major activity like planning, risk analysis, engineering and customer evaluation. The software process cycle begins with the planning activity represented by the first quadrant of this model (upper-left quadrant). Each cycle here begins with the identification of objectives for that cycle, the alternatives and constraints associated with that objective. The second quadrant is associated with risk analysis activity. It used to evaluate different alternatives that are based on the objectives and constraints listed in the first quadrant. The importance of evaluation is that, there is a risk assessment phase to evaluate the development effort and the associated risk involved for that particular iteration. The third quadrant is about engineering activity, which actually involves the development of the software, and uses various development strategies that resolve the uncertainties and risks. It may include some activities like simulation; bench marking etc. There last phase is the customer evaluation phase. It involves a review of the preceding development stage. Based on the outcome of the development step the next phase is planned. Some important characteristics of this model are:

- It uses an iterative approach and with in each iteration it introduces a phase of risk analysis to accommodate for the changing environment.
- It allows the usage of prototyping at any stage to be able to further refine requirements and therefore reduce risk element.
- It maintains a systematic approach as in the case of waterfall model.
- Although this model is quite flexible, it has some problems associated with it like; risk analysis is a major phase in this model and assessment of risks involve a greater technical expertise. And also since this approach is relatively new, there are not many practitioners in this unlike other methods.

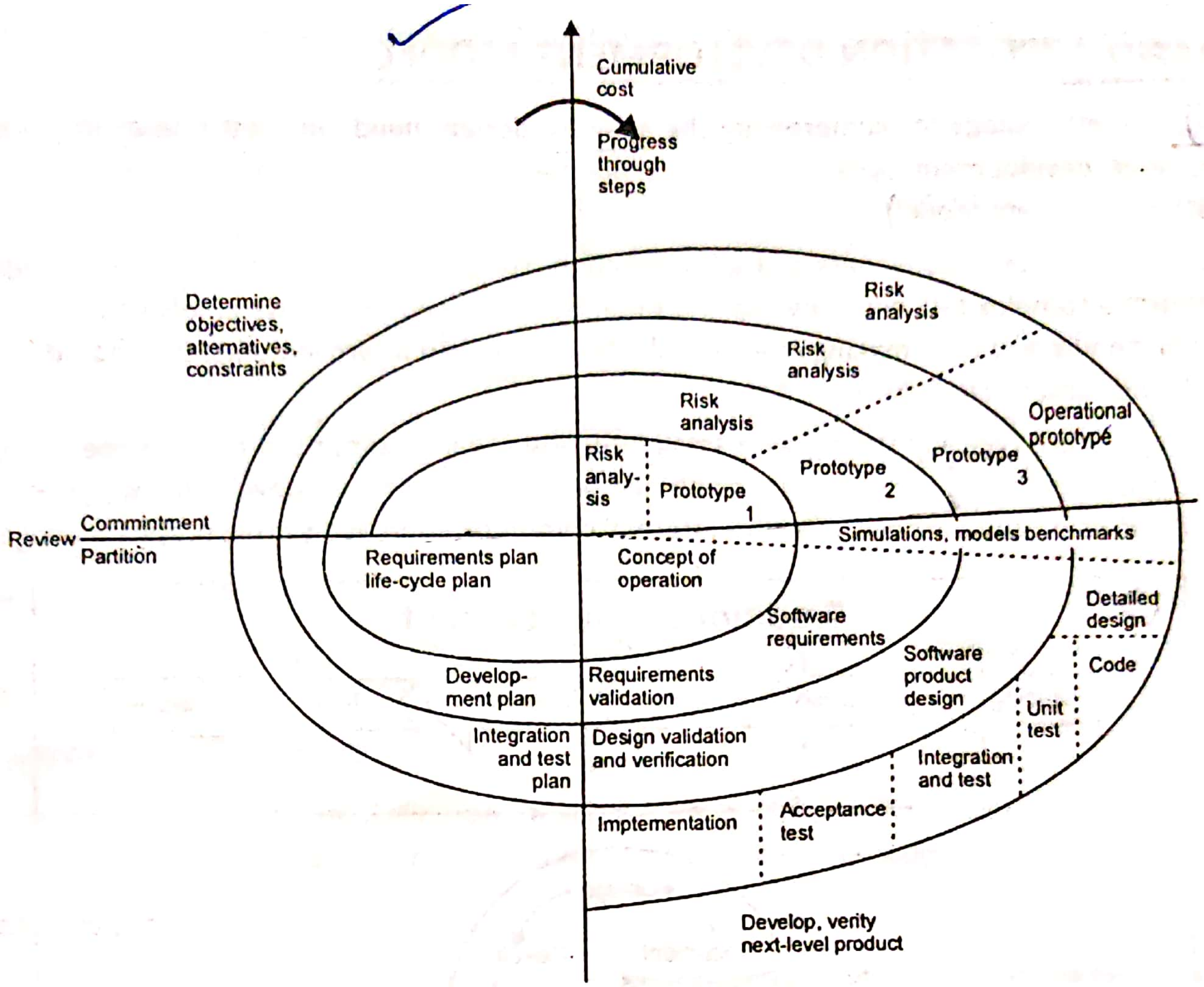


Fig. Spiral Model

Advantages

- Estimates (*i.e.,* budget, schedule, etc.) become more realistic as work progresses, because important issues are discovered earlier.
- It is more able to cope with the changes that software development generally entails.
- Software engineers can get their hands in and start working on the core of a project earlier.

Disadvantages

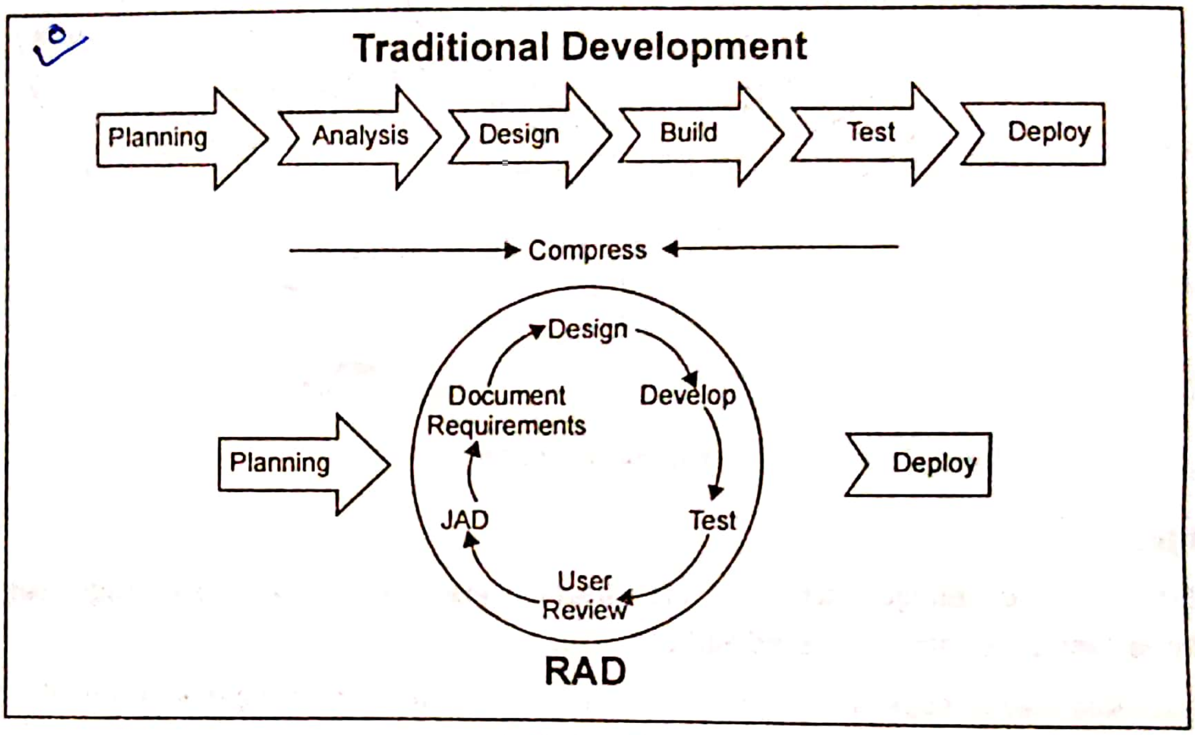
- It provides more flexibility.
- There is a lack of explicit process guidance in determining objectives, alternatives etc.
- This model is associated with iterative model, it also has the disadvantages of iterative model.
- This model is not widely used because it is relatively new.
- Great deal of reliance on risk-assessment expertise.
- Needs further elaboration of steps like parameters for cycle reviews.

2.6 RAPID APPLICATION DEVELOPMENT MODEL

1 RAD is a methodology for compressing the analysis, design, build, and test phases into a series of short, iterative development cycles. This has a number of distinct advantages over the traditional sequential development model)

Iteration allows for effectiveness and self-correction. Studies have shown that human beings almost never perform a complex task correctly the first time. However, people are extremely good at making an adequate beginning and then making many small refinements and improvements. We should encourage and exploit this rather than fight it.

RAD projects are typically staffed with small integrated teams comprised of developers, end users, and IT technical resources. Small teams, combined with short, iterative development cycles optimize speed, unity of vision and purpose, effective informal communication and simple project management.



RAD Model

An important, fundamental principle of iterative development is that each iteration delivers a functional version of the final system. It is a properly engineered, fully working portion of the final system and is not the same as a prototype. For example, the first iteration might deliver 100% of 10%, the second iteration 100% of 25%, etc.

"Rapid Application Development (RAD) is a development lifecycle designed to give much faster. Development and higher-quality results than those achieved with the traditional Lifecycle. It is designed to take the maximum advantage of powerful development. Software that has evolved recently."

Essential Aspects of RAD

Rapid Application Development has four essential aspects : *methodology, people, management and tools.* If any one of these ingredients is inadequate, development will not be high speed. Development lifecycles, which weave these ingredients together as effectively as possible, are of the utmost importance.

Methodology

The challenges facing software development organizations can be summarized as More, better, and faster. The RAD development path attacks these challenges head-on by providing a means for developing systems faster, while reducing cost and increasing quality.

Active user involvement throughout the RAD lifecycle ensures that business requirements and user expectations are clearly understood. RAD takes advantage of powerful application development tools to develop high quality applications rapidly. Prototyping is used to help users visualize and request changes to the system as it is being built, allowing applications to evolve iteratively. RAD techniques are also very. Successful when faced with unstable business requirements or when developing nontraditional Systems.

The structure of the RAD lifecycle is thus designed to ensure that developers build the systems that the users really need. This lifecycle, through the following four stages, includes all of the activities and tasks required to scope and define business requirements and design, develop, and implement the application system that supports those requirements.

➾ Requirements Planning

Also known as the Concept Definition Stage, this stage defines the business functions and data subject areas that the system will support and determines the system's scope.

➾ User Design

Also known as the Functional Design Stage, this stage uses workshops to model the system's data and processes and to build a working prototype of critical system components.

➾ Construction

Also known as the Development Stage, this stage completes the construction of the physical application system, builds the conversion system, and develops user aids and implementation work plans.

➾ Implementation

Also known as the Deployment Stage, this stage includes final user testing and training, data conversion, and the implementation of the application system.

People

The success of Rapid Application Development is contingent upon the involvement of people with the right skills and talents. Excellent tools are essential to fast application development, but they do not, by themselves, guarantee success. Fast development relies equally heavily on the people involved. These people must thus be carefully selected, highly trained, and highly motivated. They must be able to use the tools and work together in close-knit teams. Rapid development usually allows each person involved to play several different roles, so a RAD project mandates a great degree of cooperative effort among a relatively small group of people.

Each stage of a rapid development project includes activities that need to move fast.