2 Systems Development Life Cycle Models

Objectives

After completing this lesson, you should be able to do the following:

- Software Life cycle Models
- Linear Models
- Iterative Models
- Advantages and Disadvantages
- Exercises

Overview of SDLC Models

- A software lifecycle is a definition of the process which is used to develop software. It defines the phases of the development process, the order they occur in, the degree to which they may overlap, and the entry and exit criteria for each phase.
- The life-cycle models have changed over time to reflect the changes in customer group problems, perceptions of quality, and technologies available.
- They provide a solid foundation for building high quality applications.

Selection of a Model

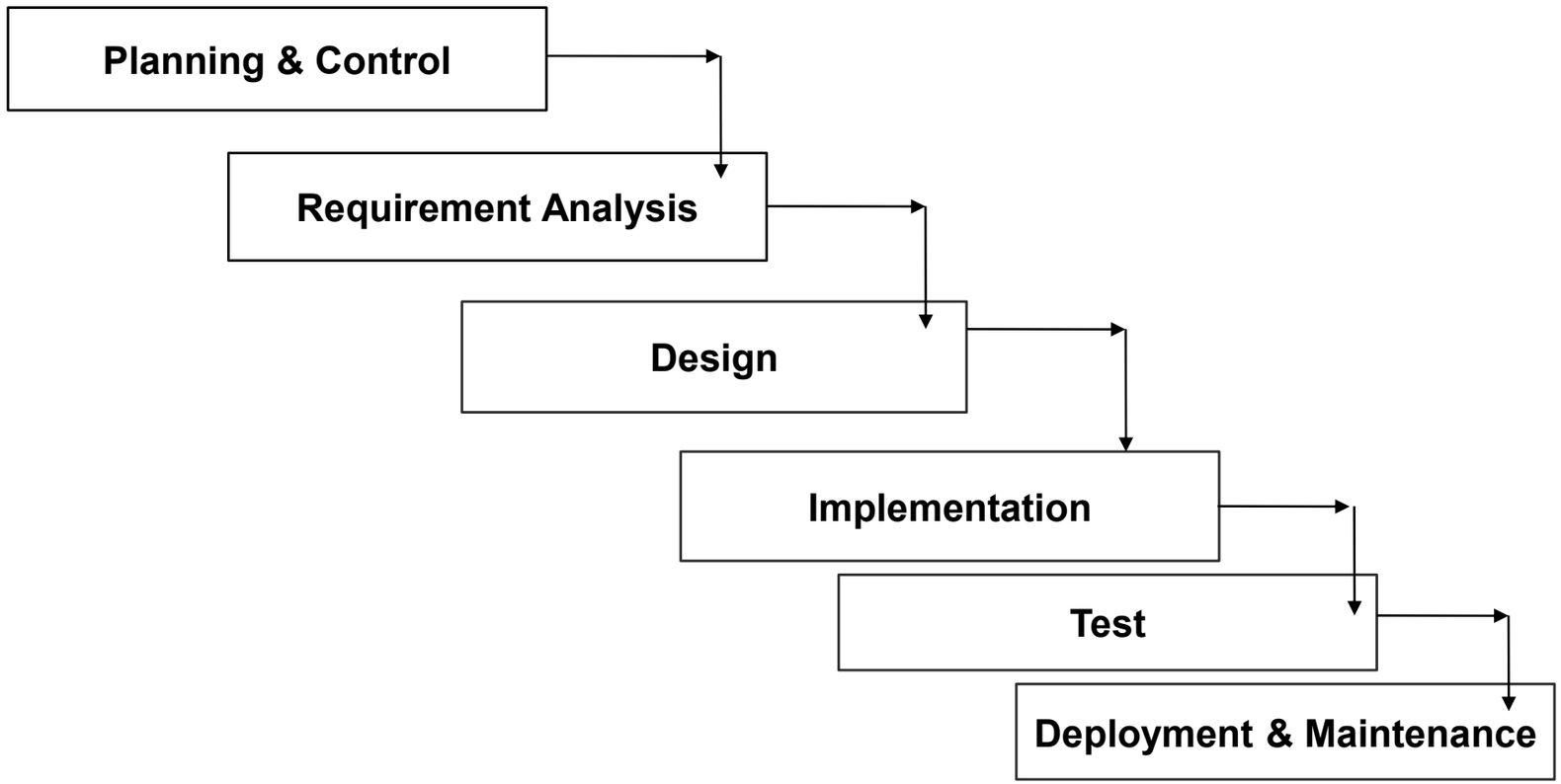
Process model is chosen based on:

- Nature of project and application domain
- Methods and tools to be used
- Controls and deliverables required
- Requirements
- Users
- Risks involved
- Budget
- Timeline

Types of Process Models

- Linear software process models:
  - Waterfall
  - V-Process
- Rapid Application Development process models:
  - Incremental Model
  - Spiral Model
- Prototype model
- Agile Models

Waterfall Model



Characteristic of Waterfall Model

- The waterfall model was the first structured software development process proposed and accepted by the software developing community.
- During the 1970's the United States Department of Defense (DoD) adopted the waterfall model as part of the American Military Standards.
- Also called sequential/linear
- Derives its name due to the cascading effect from one phase to the other phase
- Each phase is characterised by well defined starting and end point, with identifiable deliverables at each phase

Waterfall Strengths & Limitations

Strengths

- Easy to understand, easy to use
- Provides structure to inexperienced staff
- Clarity on Milestones
- well understood
- Works well when quality is more important than cost or schedule

Limitations

- All requirements must be known upfront.
- Deliverables created for each phase are considered frozen – inhibits flexibility.
- Little opportunity for customer to preview the system (until it may be too late)

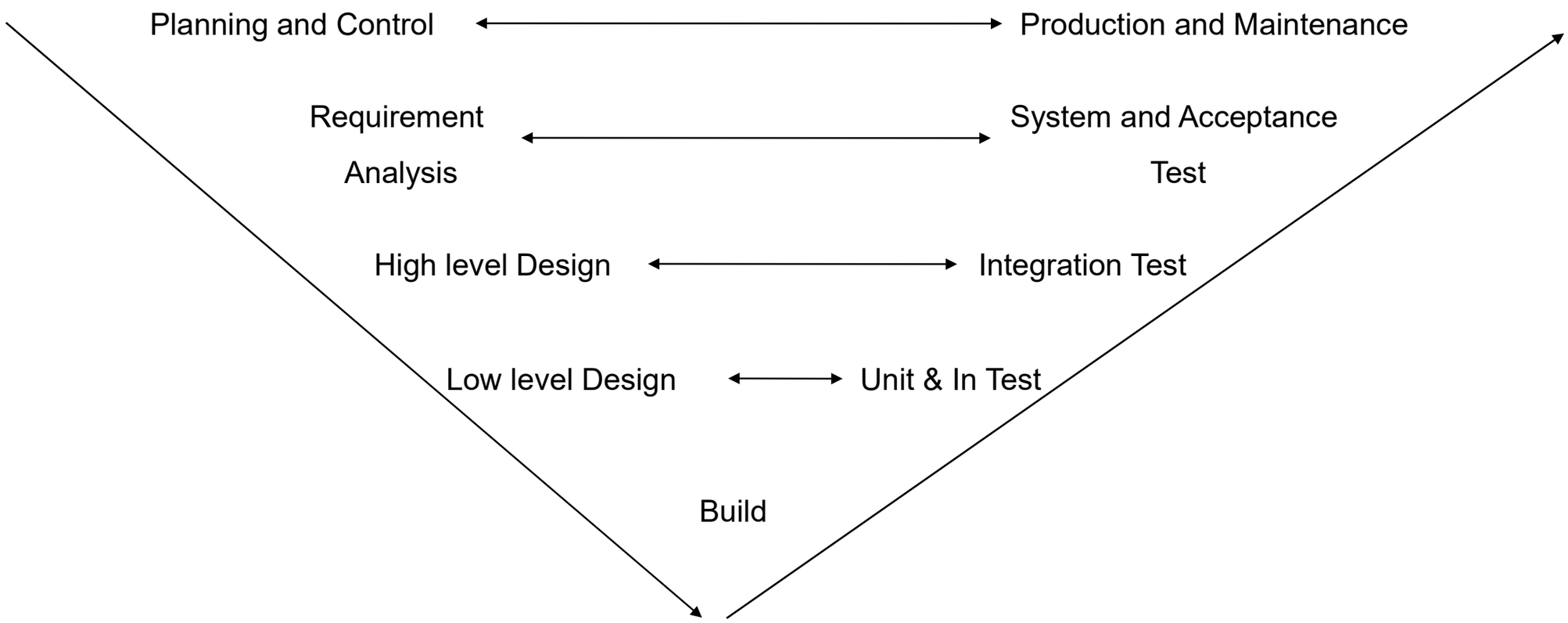
When to use Waterfall Model

- Requirements are very well known
- Product definition is stable
- Technology is understood
- New version of an existing product
- Porting an existing product to a new platform

V Model

- Refined version of waterfall model
- Greater emphasis on Testing and Quality
- V- Model provides a structured testing framework to guide work throughout the development process.
- The V model proposes an approach to software development in which both the software development process and the software test process begin simultaneously.
- To reduce the cost of correcting defects, the defects must be detected early in the development life cycle.
- Designed to support the achievement of stage containment by organizing the verification, validation in and across all of the methodology elements throughout the Delivering phase of the methodology.

V Model

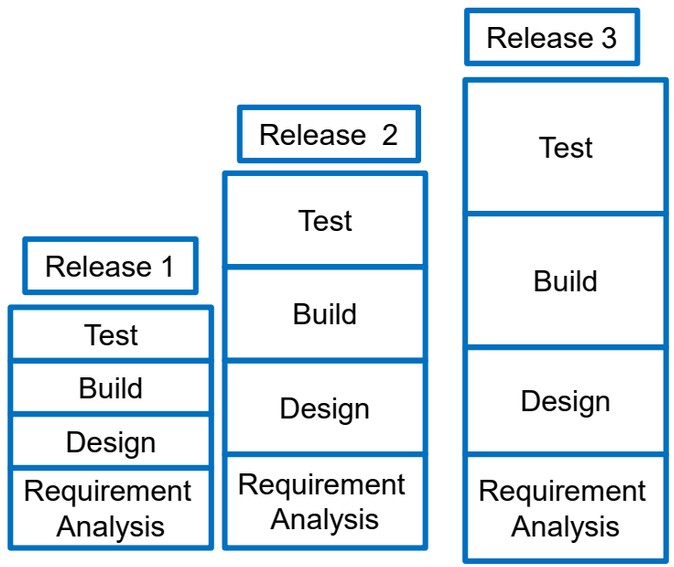

Planning and Control
Production and Maintenance
Requirement Analysis
System and Acceptance Test
High level Design
Integration Test
Low level Design
Unit & In Test
Build

Guiding Principles

The V-model framework is based upon the following Guiding principles:

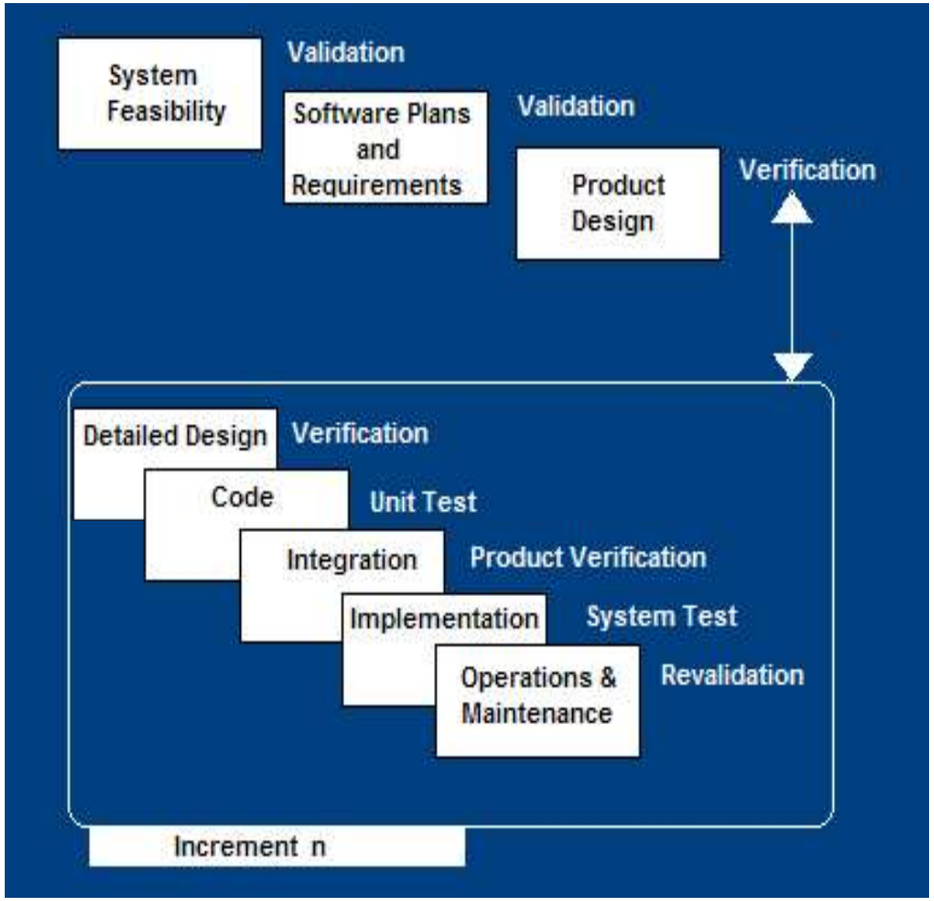
- Stage Containment
- Verification & Validation
- Entry and Exit Criteria

Incremental Model : Model



Each release delivers an operational product

Incremental SDLC Model



- Construct a partial implementation of a total system
- Then slowly add increased functionality
- The incremental model prioritizes requirements of the system and then implements them in groups.
- Each subsequent release of the system adds function to the previous release, until all designed functionality has been implemented.

Incremental Model Strengths and Weakness

Strengths:

- Develop high-risk or major functions first
- Each release delivers an operational product
- Customer can respond to each build
- Initial product delivery is faster
- Lowers initial delivery cost
- Customers get important functionality early

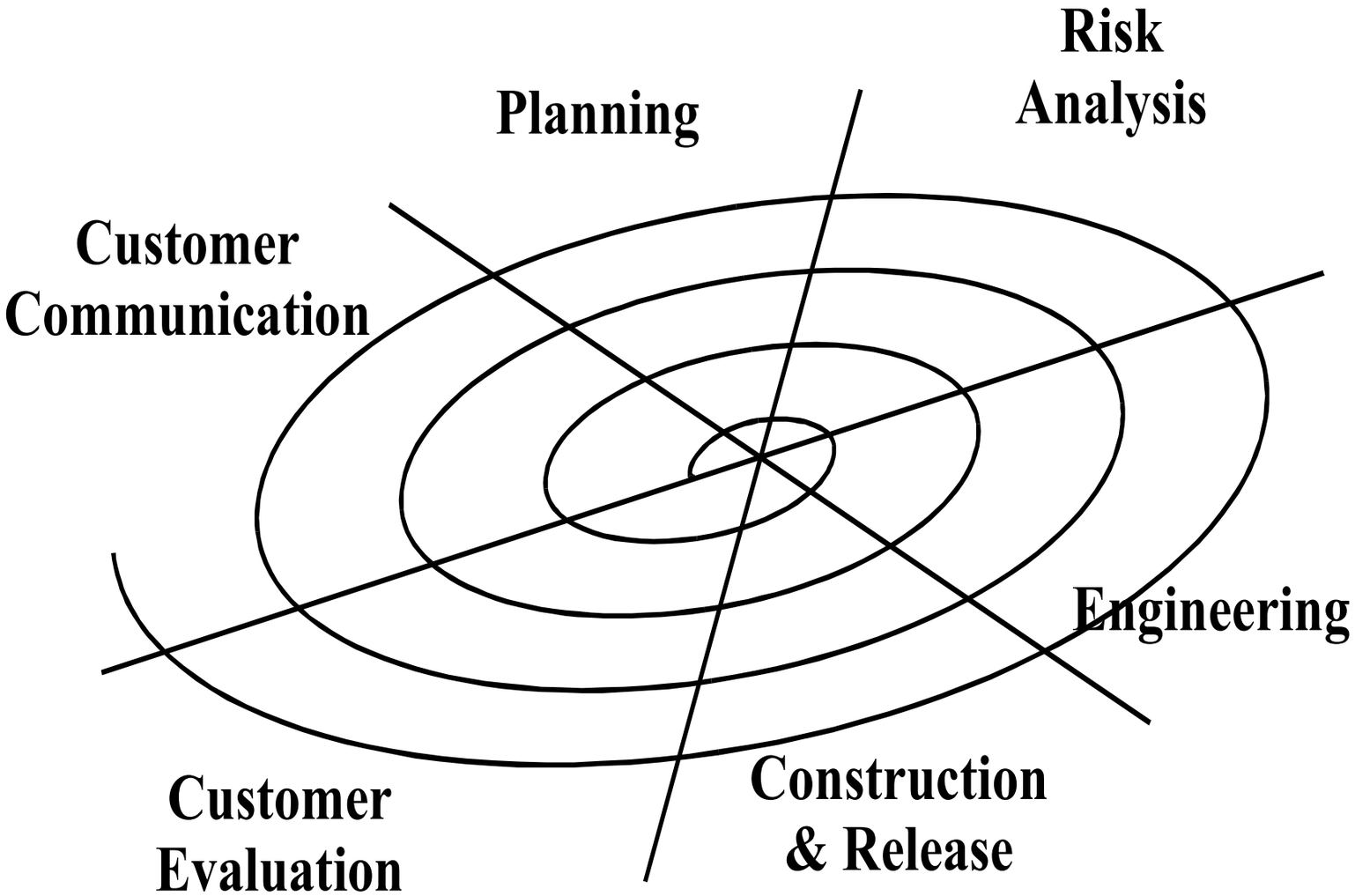
Weaknesses

- Requires good planning and design
- Requires early definition of a complete and fully functional system to allow for the definition of increments
- Total cost of the complete system is high.

When to Use Incremental Model

- Risk, funding, schedule, program complexity, or need for early realization of benefits.
- Most of the requirements are known up-front but are expected to evolve over time
- A need to get basic functionality to the market early
- On projects which have lengthy development schedules
- On a project with new technology

Spiral Model



Characteristics Of Spiral Model

- Developed by Barry Boehm in 80's
- Includes the best of classic model & prototyping in each iteration.
- Widely used model for commercial applications

Spiral Model

Each iteration consists of

- Planning and Risk analysis
- Engineering
- Customer evaluation
- Iteration continues till customer accepts the product

Spiral Quadrant – Plan Typical activities

- Develop project plan
- Develop configuration management plan
- Develop a test plan
- Develop an installation plan

Spiral Model Strengths and Weakness

Strengths:

- Facilitates the support for changes within the lifecycle
- This model provides early indication of insurmountable risks, without much cost
- Users can be closely tied to all lifecycle steps
- Users see the system early because of rapid prototyping tools
- Early and frequent feedback from users

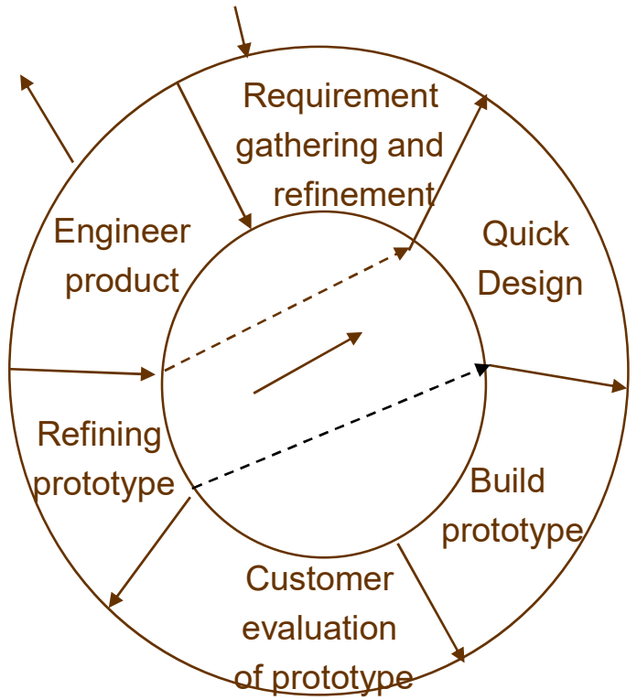
Weaknesses:

- Time spent for evaluating risks too large for small or low-risk projects: Time spent planning, resetting objectives, doing risk analysis and prototyping may be excessive
- The spiral model is more complex and harder to manage.
- This method usually increases development costs and schedule.
- Risk assessment expertise is required

When to use Spiral Model

- When creation of a prototype is appropriate
- When costs and risk evaluation is important
- For medium to high-risk projects
- Long-term project commitment unwise because of potential changes to economic priorities
- Users are unsure of their needs
- Requirements are complex
- New product line
- Significant changes are expected (research and exploration)

Evolutionary Model Prototyping



Prototype construction is standard practice in engineering

Objectives of the prototype model is to quickly develop a working model to be presented to the user so that feedbacks on the software requirements can be elicited.

Prototyping Model Strengths

- Prototype serve to reduce risk of misunderstanding customer requirements, Risk of unfamiliarity with implementation tools, Risk of unfeasible architecture
- Project can begin without fully defining or understanding requirements.
- Final requirements are improved and more in line with real user needs.
- Risks are spread over multiple software builds and controlled better.
- Operational capability is achieved earlier in the program.
- Newer technology can be incorporated into the system as it becomes available during later prototypes.
- Documentation emphasizes the final product instead of the evolution of the product.
- This method combines a formal specification with an operational prototype.

Prototyping Model Limitations

- Increase in both cost and schedule
- Management activities are increased.
- Instead of a single switch over to a new system, there is an ongoing impact to current operations.
- Prototypes change between cycles, adding a learning curve for developers and users.
- The necessary, behind-the-scenes work such as data base normalization, documentation, testing, and reviews for efficiency have not been done.
- The design of the system can sometimes suffer because the system is built in a series of “layers” without a global consideration of the integration of all other components.

Areas of Application

- Requirements are unstable or have to be clarified
- Humans are an integral part of the system.
- Have rapidly changing software technology.
- Have a large number of diverse users.
- As the requirements clarification stage of a waterfall model
- Develop user interfaces
- Short-lived demonstrations
- New, original development

RAD (Rapid Application Development)

- Rapid Application Development (RAD) is a software development methodology that focuses on building applications in a very short amount of time.
- It was originally intended to describe a process of development that involves application prototyping and iterative development
- Hybrid of prototyping, jad, case
- Produce system in 6 months or less

RAD Advantages

- RAD reduces the development time
- Reusability of components helps to speed up development
- Ability to rapidly change to system design at user's request

RAD Disadvantages

- Quality may be sacrificed for speed
- Time consuming for key personnel
- RAD is based on OO approach, hence may not work well for all projects
- Possible shortcuts on internal standards, module reusability
- Might allow less time to develop quality, consistency and design standards

When to Use RAD

- Reasonably well-known requirements
- User involved throughout the life cycle
- Project can be time-boxed
- Functionality delivered in increments
- High performance not required
- Low technical risks
- System can be modularized

Exercise Define a Proper SDLC Models for the below mentioned scenarios

1. To develop a simple text editor
2. An online inventory management system for an automobile industry.
3. A new system for comparing finger prints. It is not clear if the current algorithm can compare finger prints in the given response time constraints.
4. A spreadsheet system which has some basic features and many other desirable features that use these basic features.
5. The exercise describes a software development project called Gtrials. It is an existing world class pharmaceutical project management software product running on windows7 developed in Java. Functionally the product is perfect, but has issues with performance. The product team has decided to change the design and redo certain features using struts .The team is not sure how the new design and logic will work .
6. Knowledge Buzz" library has a huge collection of books and has branches all over India. Lending of Books , membership creation, withdrawal of membership, etc is done manually at present. "Knowledge Buzz" requires a software which could be accessed by any of the clients in India, membership should be created online or manual, scanning of books be done online, booking of books should be done online, withdrawal of membership should be done online.

Solution : Exercise

1. Waterfall
2. RAD model
3. Evolutionary Prototype
4. Incremental
5. Evolutionary Prototype
6. Spiral

Summary

- SDLC Models
- Characteristics of Models
- Strengths and limitations
- Applicability of the models
- Exercise and solutions