UNIT 2 :
SOFTWARE DEVELOPMENT MODEL

- A **software process model** is a simplified representation of a software process. Each model represents a process from a specific perspective.
- These **generic models** are abstractions of the process that can be used to explain different approaches to the software development.
- In Software process discussed above are common activities in every model **Communication, Modeling, Planning and Constructions.**

- In software Engineering number of software development life cycle models are available.
- Each model has a particular life cycle in order to ensure success in process of development.
- **Software Development Life Cycle** models.

    (1) Waterfall model

    (2) Prototype model

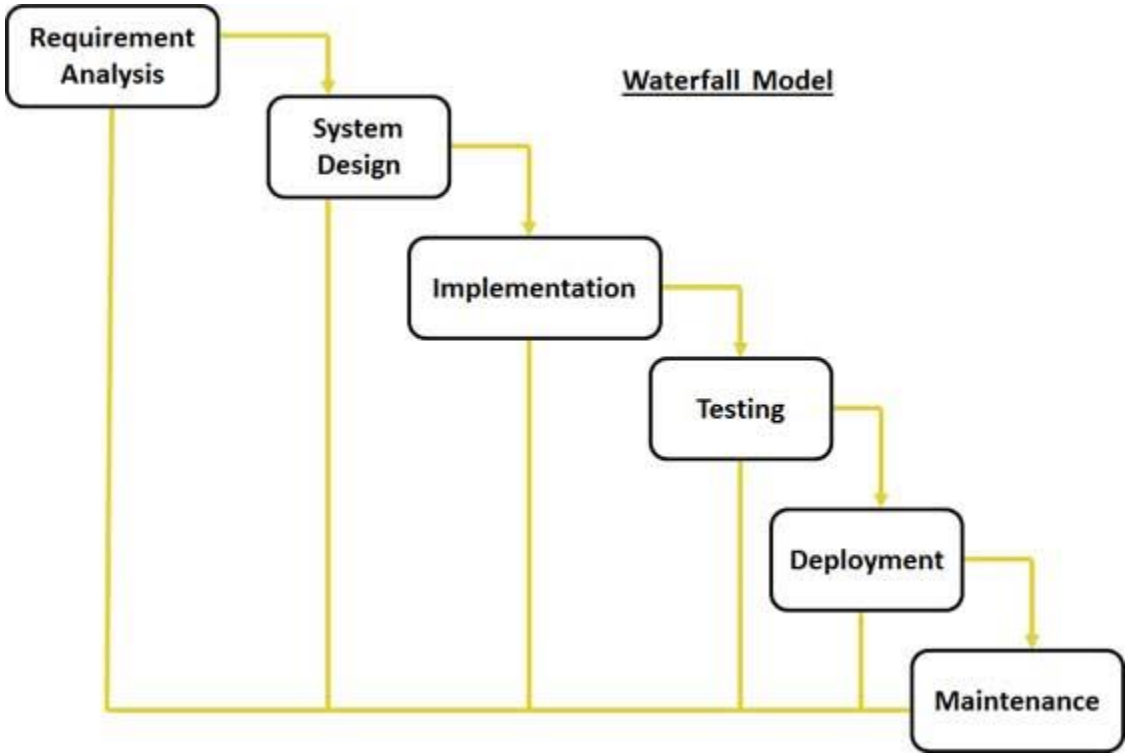
    (3) Spiral model

    (4) Rapid Application Development model.

2.1 Waterfall Model:

- Waterfall model is the first software development model to be used generally in software engineering.
- also known as Linear Sequential Life Cycle Model.
- It is very simple to understand and easy to use.
- In waterfall model, phases must be complete sequentially so that phase completed full and then next phase begin.
- Waterfall model provide systematic and sequential approach of software development that begins from requirement gathering and improvement through design, implementation, testing, deployment and maintenance.

Waterfall Model



Phases of Waterfall Model

- **Requirement Analysis:**
- This is the first phase of waterfall model which start with understand the requirements from customers so that software meets his specifications. In this phase requirements are collected, analyzed and then prepare proper documentation which helps further development process.

System design:

- First design the Hardware and Software as per the requirements of every part of system.
- Algorithms and diagrams defining the scope and objectives of each logical model are developed.
- Then with the help of logical representative technique, design the logical modules for each process of system.

System Implementation:

- This is the important phase of software development process in which actual coding is started.
- A software program is written base upon the algorithm designed in the system design phase.
- Each unit is developed and then tested for its functionality.
- This verifies the expectation of user from particular unit.

System Testing:

- In the system testing a series of test and test cases are considered to check for errors, faults and bugs.
- The various tools and software are used for testing in order to make sure it is error free.

System deployment and maintenance:

- This is the last phase of waterfall model. Once the testing is done then system is deployed and ready to use or deployed in customer environment.
- After the software has been deployed it is the duty of system analyst provides maintenance to customer while problem comes during running the system or problem comes from any other reasons.

Advantages and Disadvantages of Waterfall Model :

- **Advantages:**
- Waterfall model is very simple and easy to understand and use.
- Steps execution is sequential and completed one at a time so that no chance to forget some process or part of system for development.
- If requirements specification is clear, the small project can be well developed.
- Each step has a specific a review process so that no chances for mistake.
- Results and process are well documented.

Disadvantages:

- It is often difficult for the customer to state all requirements.
- Maximum chances of hazard and doubt.
- Poor model for long and ongoing project and not a good for complex and object oriented projects.
- Difficult to integrate risk management.

2.2 Prototype Model:

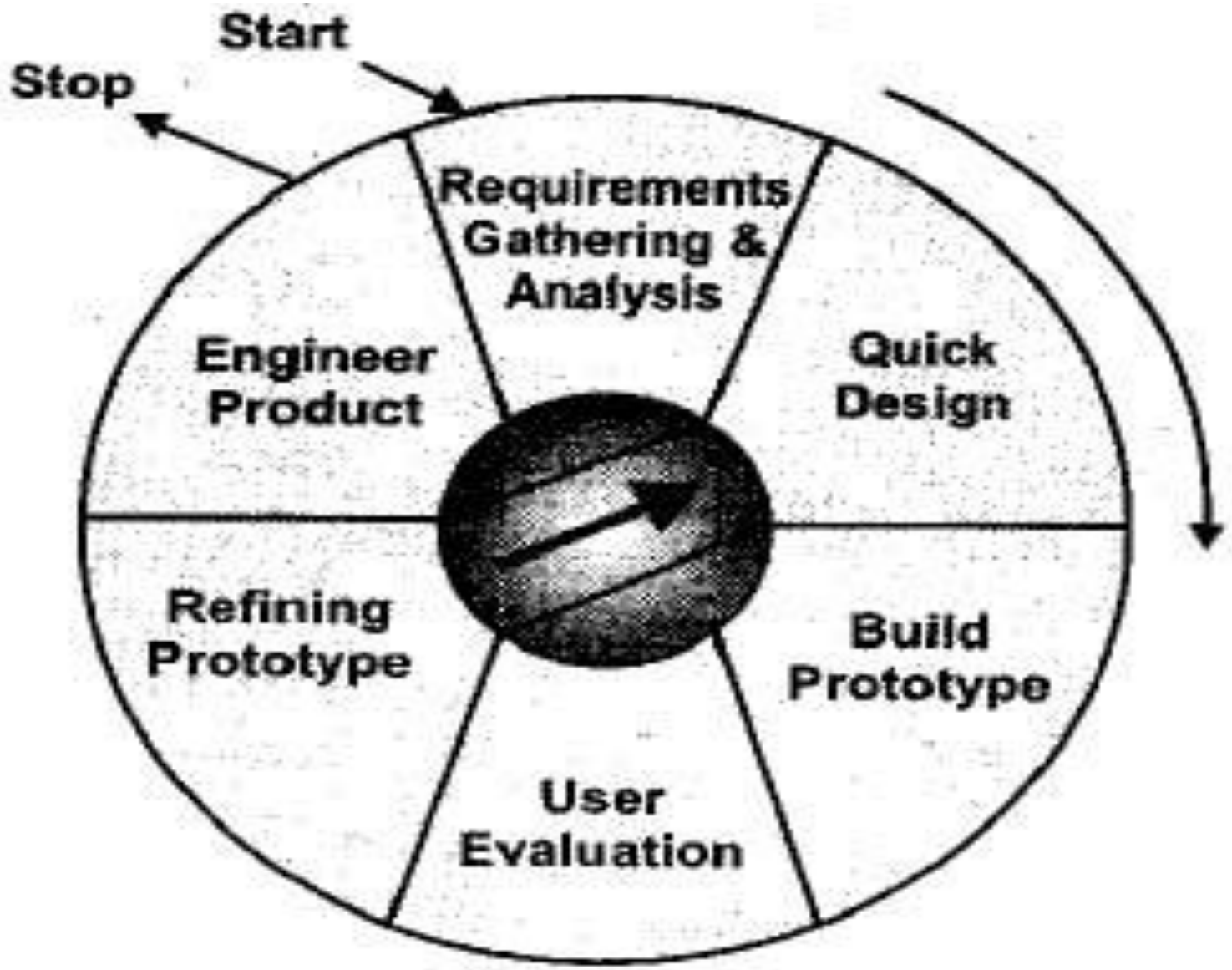
As a point of view of software development there are two major problems to system development

- **(1) Software development life cycle take too long**
- **(2) The correct system developed first time only.**

Long processes frustrate the user so that the alternative advance software development technique is Prototyping.

- The software prototyping is becoming very popular model for software development.
- In this model at early stage to understand the customer requirement and developers understand what customer exactly wants.
- Then an early approximation of a final prototype is built, tested and reworked as necessary until an acceptable prototype is finally achieved from which the complete system can now be developed.
- The most expensive research on prototyping model has been conducted by Naumann and Jenkins

Prototype Model:



Steps of Prototype Model:

- **Requirement gathering and analysis:**
- This is the first step of prototype model.
- In this developers collect the required information of existing system is detail by using various fact finding techniques.
- This usually involves questionnaire, survey, review of documents and interviewing methods.
- After collecting information from various sources then analyze it and determine specific information for new system.

Quick Design:

- Quick design helps in developing prototype.
- When requirements are satisfied, a preliminary design or quick design is created.
- It is not detail design but includes on the important aspect of the system which an idea of the system to the user.

- **Build prototype:**

- From the preliminary design a first prototype of new system is constructed which represent the working model of the required system.

User Evaluation:

- Evaluate the first prototype by user and give suggestions or changes to developers
- i.e. what is to be added or what is to be removed?
- Then developers analyze the user's suggestions or changes.

Refining prototype:

- According to the suggestions of user, developers refine the first prototype.
- This is a new prototype including the suggestions. Once again it will evaluate by user. This process will be continuing until users are satisfied.
- Then second prototype is developed and evaluated in the same way as was the first prototype.
- The earlier steps are iterated as many times as necessary until the users are satisfied that the prototype represents the final product desired. Then final system is constructed.

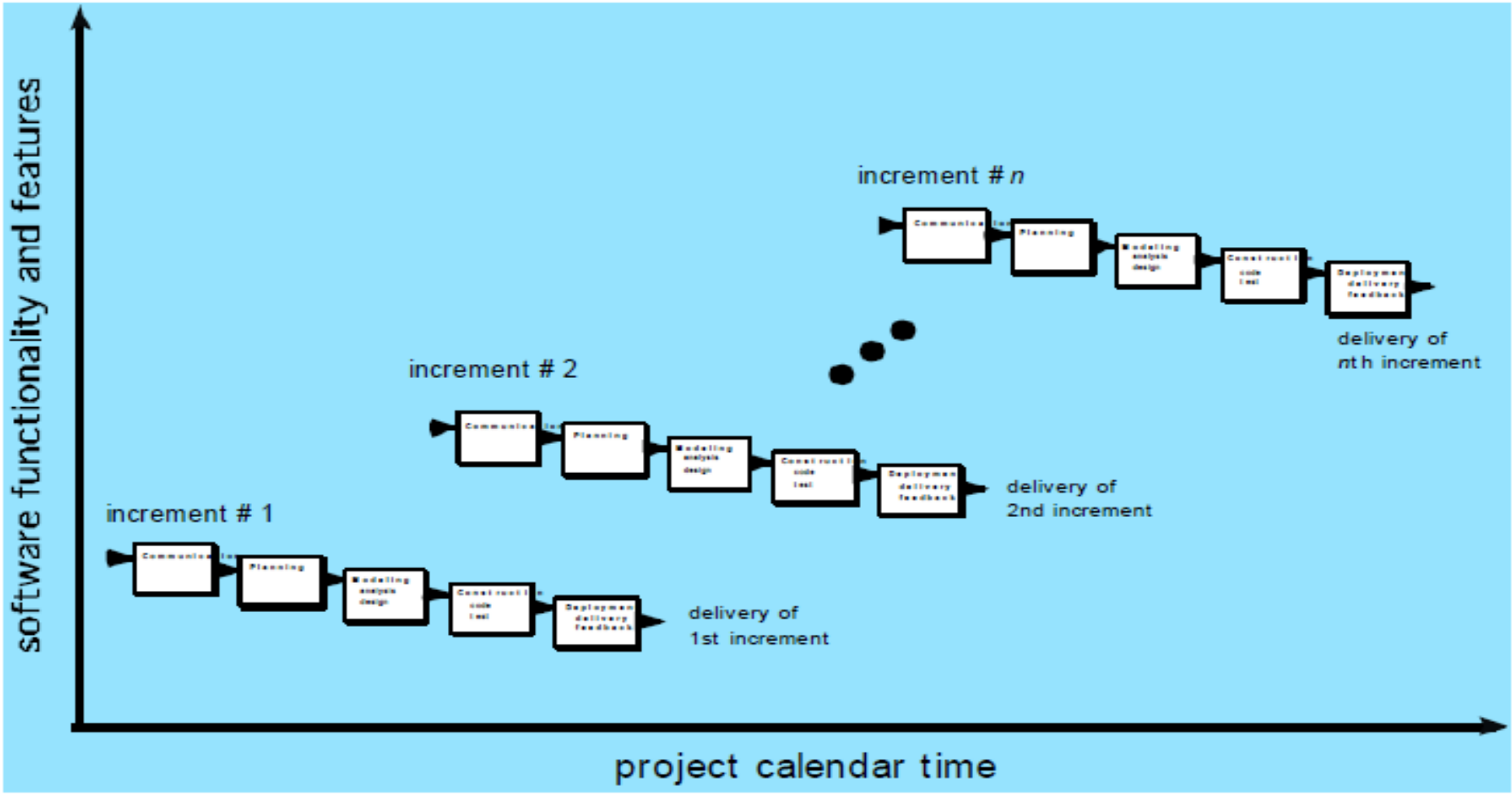
- **Engineer Product:**
- This is the last step of prototype model. In this user accept the final prototype. The final system is evaluated thoroughly followed by the routine on regular basis for preventing large-scale failures and minimize downtime.

Advantages and disadvantages:

- **Advantages:**
- Reduce time and costs: Prototyping model can improve quality of requirements and specifications provided to developers. Because changes cost exponentially more to implement as they are detected later in development the early determination of what the user really wants can result in faster and less expensive software.
- Improve user involvement: Prototyping requires user involvement and allows them to see and interact with a prototype allowing them to provide better and more complete feedback.
- Every prototype evaluate by user so that customer is satisfied from the system.
- Missing function can be identified easily.
- Helps in reducing risks associated with the software.
- It is faster than waterfall and SDLC approach.

- **Disadvantages:**
- Users may get confused in the prototype and actual system.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- If the user is not satisfied by the developed prototype, then a new prototype is developed. This process goes on until a perfect prototype is developed. Thus, this model is time consuming and expensive.

2.3 Incremental Process Model:



- **Incremental Process Model** most widely used effectively in the industry.

- The *incremental model* combines elements of the *waterfall* model applied in an iterative fashion. The model applies linear sequences in a staggered fashion as calendar time progresses.

- Each linear sequence produces deliverable “increments” of the software. (Ex: a Word Processor delivers basic file mgmt., editing, in the first increment; more sophisticated editing, document production capabilities in the 2nd increment; spelling and grammar checking in the 3rdincrement.

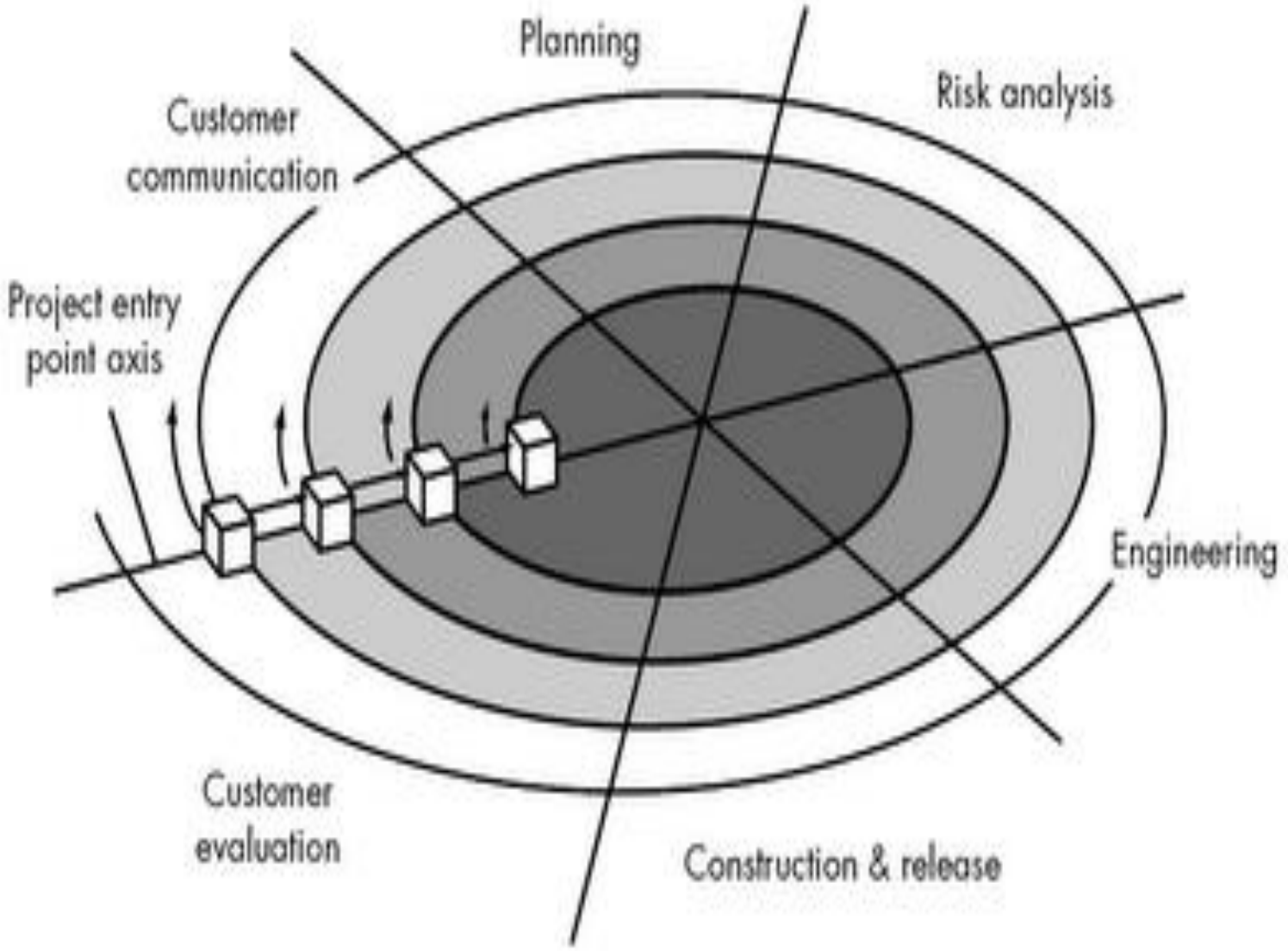
- When an increment model is used, the 1st increment is often a *core product*. The core product is used by the customer.

- As a result of use and / or evaluation, a plan is developed for the next increment.
- The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality.
- The process is repeated following the delivery of each increment, until the complete product is produced.
- If the customer demands delivery by a date that is impossible to meet, suggest delivering one or more increments by that date and the rest of the Software later.

Spiral Model:

- This model was introduced by Barry Boehm in his 1986 research paper " A Spiral model of software development and enhancement".
- Spiral Model is an evolutionary software process model which is a combination of **logical part of waterfall model and iterative character of prototyping.**
- This model more emphasis on risk analysis.

Spiral Model



- A spiral model is divided into six tasks (also known as framework activities)
- Tasks (framework activities) includes:
  - customer communication,
  - Planning
  - risk analysis
  - construction
  - release
  - customer evaluation.

- Every task represents one section of the spiral path.
- As development process starts, the software team performs activities that are indirect by a path around the spiral in a clockwise direction.
- Development process begins at the center of spiral model.

Steps of Spiral Model:

Customer Communication:

- **Customer Communication** is the very important task of spiral model.
- The task required to establish the effective communication between developer and customer.
  - ❑ Actually what customer wants to develop?
  - ❑ What are the expectations?
- Effective communication provides better output.
- If sufficient information is available from customer, systematic processing can be done by developer which helps to develop high-quality system.

Planning:

- **Planning** includes defining resources, timelines and other project related information.
- A developer has prepared planning for overall system.
  - That is in one spiral
    - which part is complete
    - what are the contents of each task?
    - What are the resources of each task?
    - What is execution time for each task?
- They are going through planning which helps to avoid errors or faults.

Risk Analysis:

- **Risk Analysis t**ask includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overrun.

- The main purpose of risk analysis is

    - to understand risks in enhanced ways
    - to verify and correct attributes
    - to provide alternative solutions.

Engineering:

- According to the planning developer develop the applications which are expected by user.

Construction and release:

- In **Construction and release** Task developers construct the application, test it using various testing technique, if there is no error install the application and release to user for evaluation.

Customer Evaluation:

- After construction the application it will evaluate by user by using various methods and techniques.
- Method may be manual or automatic and obtains feedback from customer which is based on evolution of software representations created during the engineering state.
- Once user is satisfied from the result it will be launch on the user environment.

Advantages and Disadvantages:

Advantages:

- Budget and schedule becomes more realistic as work progresses.
- Especially risk analysis is available. So that developers avoid the faults with alternative solutions.
- Customer is satisfied from the process because customer feedback is there.
- Everything is in planning.
- Requirements can be captured more accurately.
- Development can be divided into smaller parts.

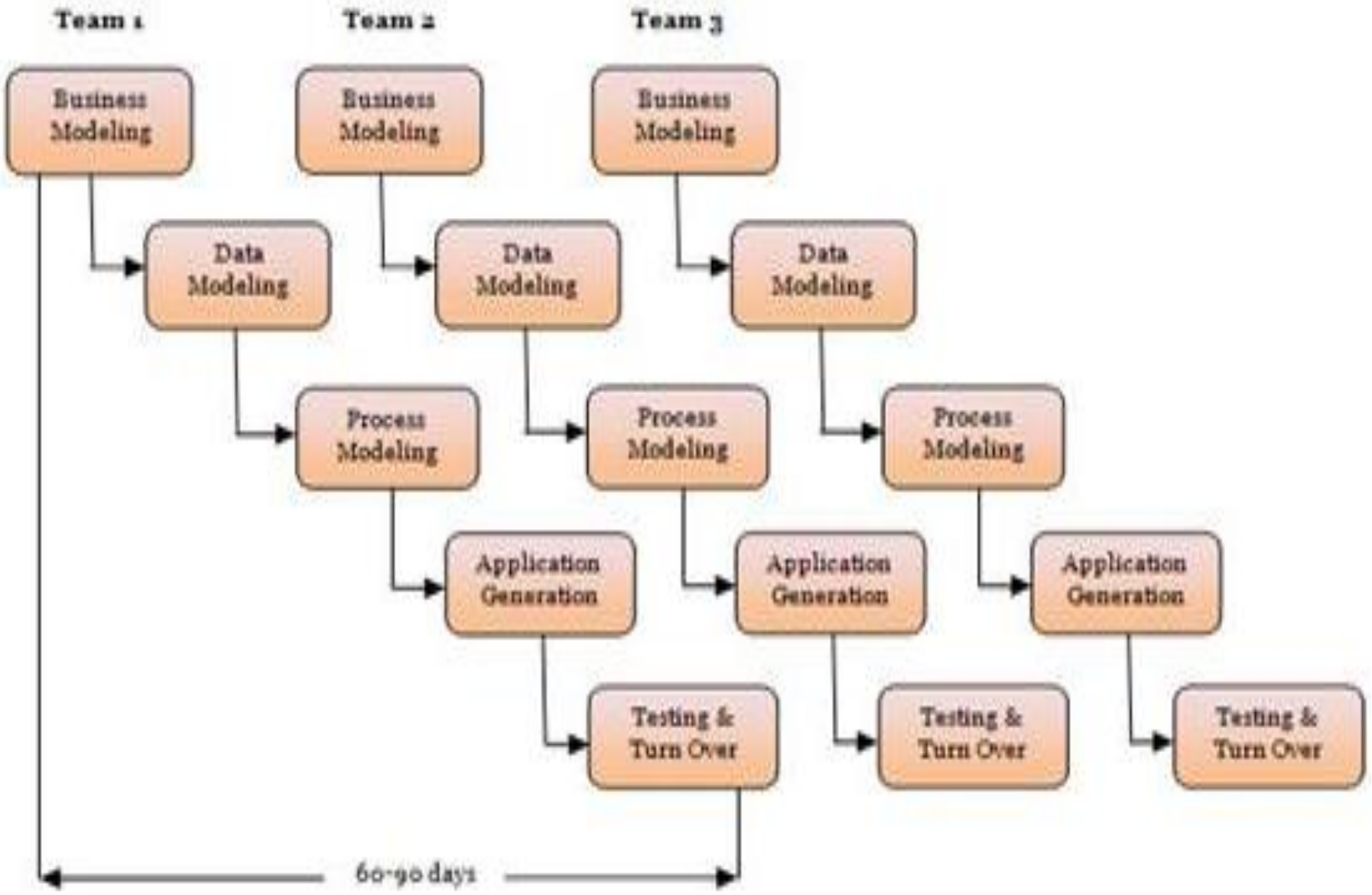
Disadvantages:

- Management is more multifaceted.
- It could be expensive for small projects
- Spiral may go indefinitely.
- Very Much documentation is required than other methods.

2.5 Rapid Application Development(RAD) Model:

- Rapid Application Development (RAD) Model is a linear sequential software process model. It is very much faster and high quality result model than traditional development model. In the Rapid Application Development (RAD) Model once requirements and projects scope and limitations are well defining, the RAD process enables a development team to create a fully functional system within very short period i.e. within 60 to 90 days. That is each major function can be addressed by the separate RAD team and finally it integrated to form a whole. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

Rapid Application Development (RAD) Model



Phases of RAD Model:

Business modeling:

- The information flow is identified between various business functions.
- A complete business analysis is performed to find important information for business in ways that answer the following questions:
    - How does information obtain?
    - What information obtained?
    - Who obtain it?
    - Where use the information?
    - Who processes the information?

Data modeling:

- The information obtained in the above phase is reviewed and analyze to form set of data object which is important for the business.

- To defining the entity/object and its attributes for data modeling and also set the relation between various entities/objects.

Process Modeling:

- In data modeling phase data objects are defining that are converted to achieve the business information flow to achieve some specific business objective.

- Process descriptions are created for adding, modifying, deleting or retrieving a data object.

Application Generation:

- Actual coding is developed and system is built by using automation tool such as VB, VC++ etc. to convert the process and data models into actual prototype.

Testing and Turnover:

- Since the RAD model process highlighted on reuse.
- That is many of the program components have already been testing.
- It reduces the risk of any major issues.
- However new components must be tested and all interfaces must be fully exercised.

Advantages and disadvantages:

- **Advantages:**
- Each major function can be addressed by the separate RAD team and finally it integrated to form a whole. So that It reduces the time.
- Iterative time can be short with use of powerful RAD tools. So that It is faster than all other system development models.
- Increases reusability component.
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

Disadvantages:

- Sufficient human resources that are the right number of RAD teams are required for large project.
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.
- Management complexity is more.
- Suitable for systems that are component based and scalable.