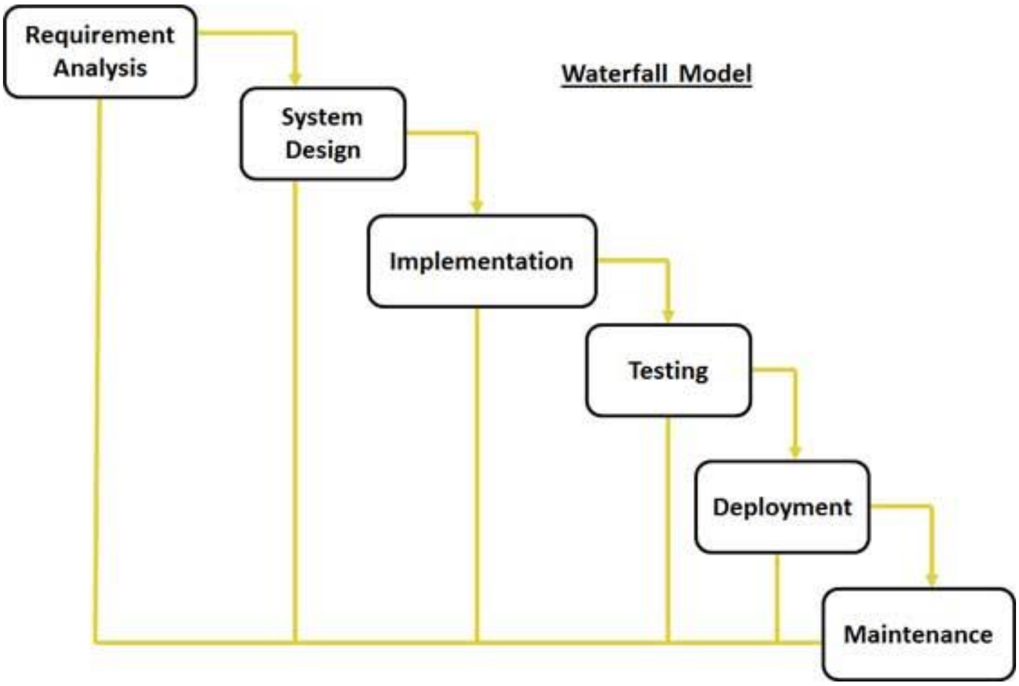
PRACTICAL 2

Aim: Introduction to Waterfall model.

Waterfall Model:

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.



The waterfall development model originates in the manufacturing and construction industries; highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Since no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development.

Waterfall Model:

In Royce's original waterfall model, the following phases are followed in order:

1. Requirements specification resulting in the product requirements document
2. Design resulting in the software architecture
3. Construction (implementation or coding) resulting in the actual software
4. Integration
5. Testing and debugging
6. Installation
7. Maintenance

Thus the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified. Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process. These variations included returning to the previous cycle after flaws were found downstream or returning all the way to the design phase if downstream phases deemed insufficient.

Advantages of Waterfall Model:

1) Waterfall model is simple to implement and also the amount of resources required for it are minimal.

2) In this model, output is generated after each stage (as seen before), therefore it has high visibility. The client and project manager gets a feel that there is considerable progress. Here it is important to note that in any project psychological factors also play an important role.

3) Project management, both at internal level and client's level, is easy again because of visible outputs after each phase. Deadlines can be set for the completion of each phase and evaluation can be done from time to time, to check if project is going as per milestones.

4) This methodology is significantly better than the haphazard approach to develop software. It provides a template into which methods of analysis, design, coding, testing and maintenance can be placed.

5) This methodology is preferred in projects where quality is more important as compared to schedule or cost.

Disadvantages of Waterfall Model:

1) Real projects rarely follow the sequential flow and iterations in this model are handled indirectly. These changes can cause confusion as the project proceeds.

2) It is often difficult to get customer requirements explicitly. Thus specifications can't be freezed. If that case arises baseline approach is followed, wherein output of one phase is carried forward to next phase. For example, even if SRS is not well defined and requirements can't be freezed, still design starts. Now if any changes are made in SRS then formal procedure is followed to put those changes in baseline document.

3) In this model we freeze software and hardware. But as technology changes at a rapid pace, such freezing is not advisable especially in long-term projects.

4) This method is especially bad in case client is not IT-literate as getting specifications from such a person is tough. Even a small change in any previous stage can cause big problem for subsequent phases as all phases are dependent on each-other.

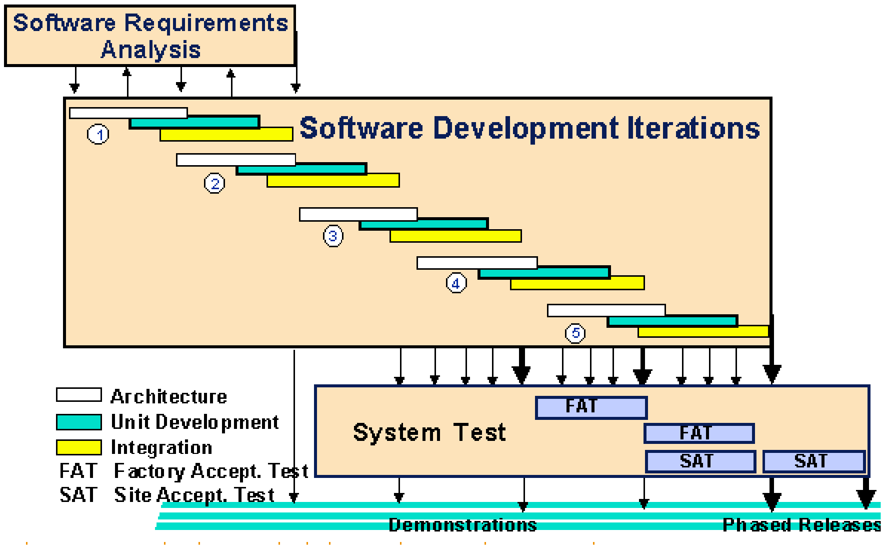
Comparison with Other Models:

A Programming process model is an abstract representation to describe the process from a particular perspective. There are numbers of general models for software processes, like: Waterfall model, Evolutionary development, Formal systems development and Reuse based development, etc. This research will view the
Following five models:

1. Iteration model.
2. Spiral model.
3. Extreme model.

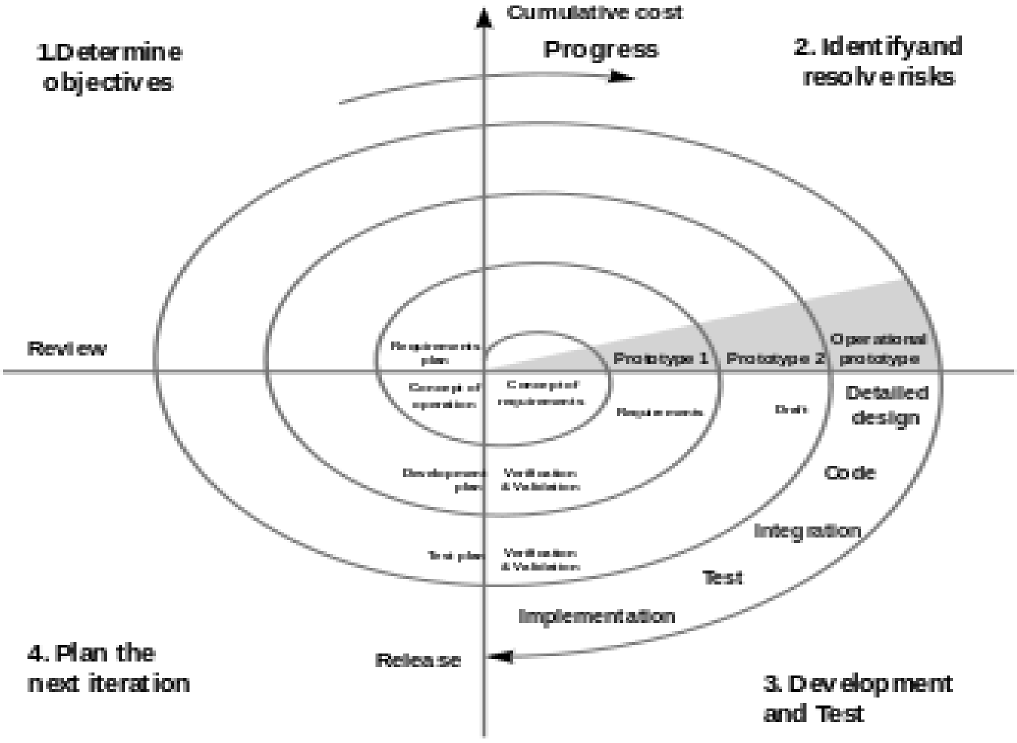
1. Iteration model.

- The problems with the Waterfall Model created a demand for a new method of developing systems which could provide faster results, require less up-front information, and offer greater flexibility.
- With Iterative Development, the project is divided into small parts. This allows the development team to demonstrate results earlier on in the process and obtain valuable feedback from system users.
- Often, each iteration is actually a mini-Waterfall process with the feedback from one phase providing vital information for the design of the next phase.
- In a variation of this model, the software products, which are produced at the end of each step (or series of steps), can go into production immediately as incremental releases.



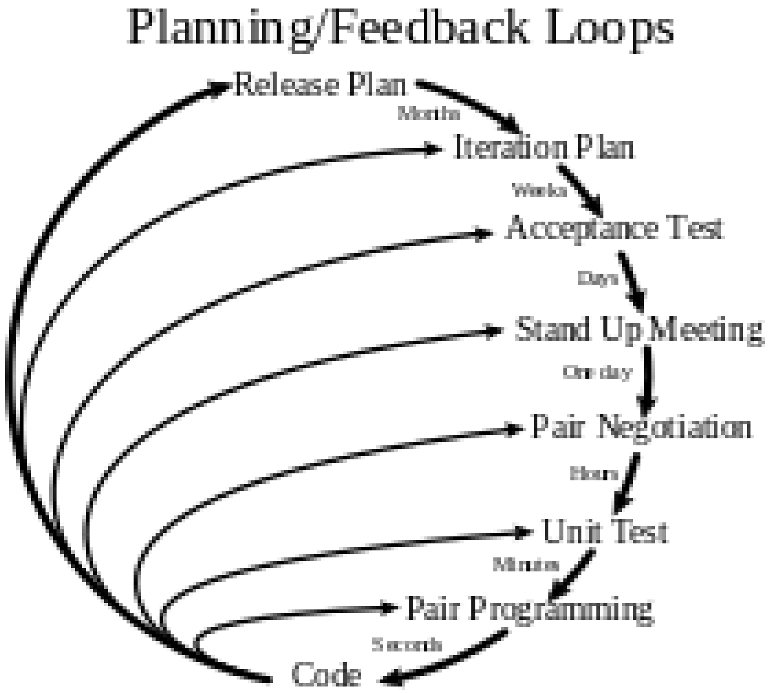
2. Spiral Model.

- The spiral model is similar to the incremental model, with more emphases placed on risk analysis.
- The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation.
- A software project repeatedly passes through these phases in iterations (called Spirals in this model).
- The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed.
- Each subsequent spiral builds on the baseline spiral. Requirements are gathered during the planning phase.
- In the risk analysis phase, a process is undertaken to identify risk and alternate solutions.
- A prototype is produced at the end of the risk analysis phase. Software is produced in the engineering phase, along with testing at the end of the phase.
- The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.



4. Extreme Programming Model:

- An approach to development, based on the development and delivery of very small increments of functionality.
- It relies on constant code improvement, user involvement in the development team and pair wise programming.
- It can be difficult to keep the interest of customers who are involved in the process. Team members may be unsuited to the intense involvement that characterizes agile
- Methods.
- Prioritizing changes can be difficult where there are multiple stakeholders.
- Maintaining simplicity requires extra work.
- Contracts may be a problem as with other approaches to iterative development.



Conclusion:

After completing this research , it is concluded that :
1. There are many existing models for developing systems for different sizes of projects and requirements.
2. These models were established between 1970 and1999.
3. Waterfall model and spiral model are used commonly in developing systems.
4. Each model has advantages and disadvantages for the development of systems, so each model tries to eliminate the disadvantages of the previous model.

Finally, some topics can be suggested for future works:

1. Suggesting a model to simulate advantages that are found in different models to software process management.

2. Making a comparison between the suggested model and the previous software processes management models.
3. Applying the suggested model to many projects to ensure of its suitability and documentation to explain its mechanical work.