Introduction to Software Engineering

As per CBSE curriculum
Class 12

Chapter- 05

By-
Neha Tyagi
PGT (CS)
KV 5 Jaipur(II Shift)
Jaipur Region

What is Software Engineering?

Software Engineering is a structured, systematic approach for the design, development and maintenance of software system.

Need for Software Engineering-

Now a days applications are heavily depend on software and modern days software are large and complex. It is very important that the software being developed-

a. Conforms to the specification.
b. Is delivered in time.
c. Works as intended and solves the problem for which it is developed.
d. Conforms to the budget i.e., costs are within budgets nearly.

Thus, Software Engineering is very important for the development of software because of the following reasons-

i. Correct specification.
ii. Scalability scope.
iii. Cost control.
iv. Quality.

Software Process Activities-

The term software process refers to a set of logically related activities, which are carried out in a systematic order that leads to the production of the software to be delivered.

There are some fundamental activities that are common to all software processes, known as process activities. These are-

1. **Software specification-** This activity is responsible for defining-
    a. The main functionality of the software.
    b. Constraints on its operation.
2. **Software Design and Implementation-**This activity is responsible for-
    a. The design of the proposed software as per software specifications.
    b. Programming as per design.
3. **Software Verification and Validation-**This activity is responsible for ensuring that-
    a. Software conforms to all the specification.
    b. The software works as per the proposed design.
4. **Software Evolution(software maintenance)-**This activity ensures two important things-
    a. The developed software meets the customer requirements.
    b. The software design ensures adaptability and scalability i.e., it evolve to meet changing customer needs.

Software Process Model-

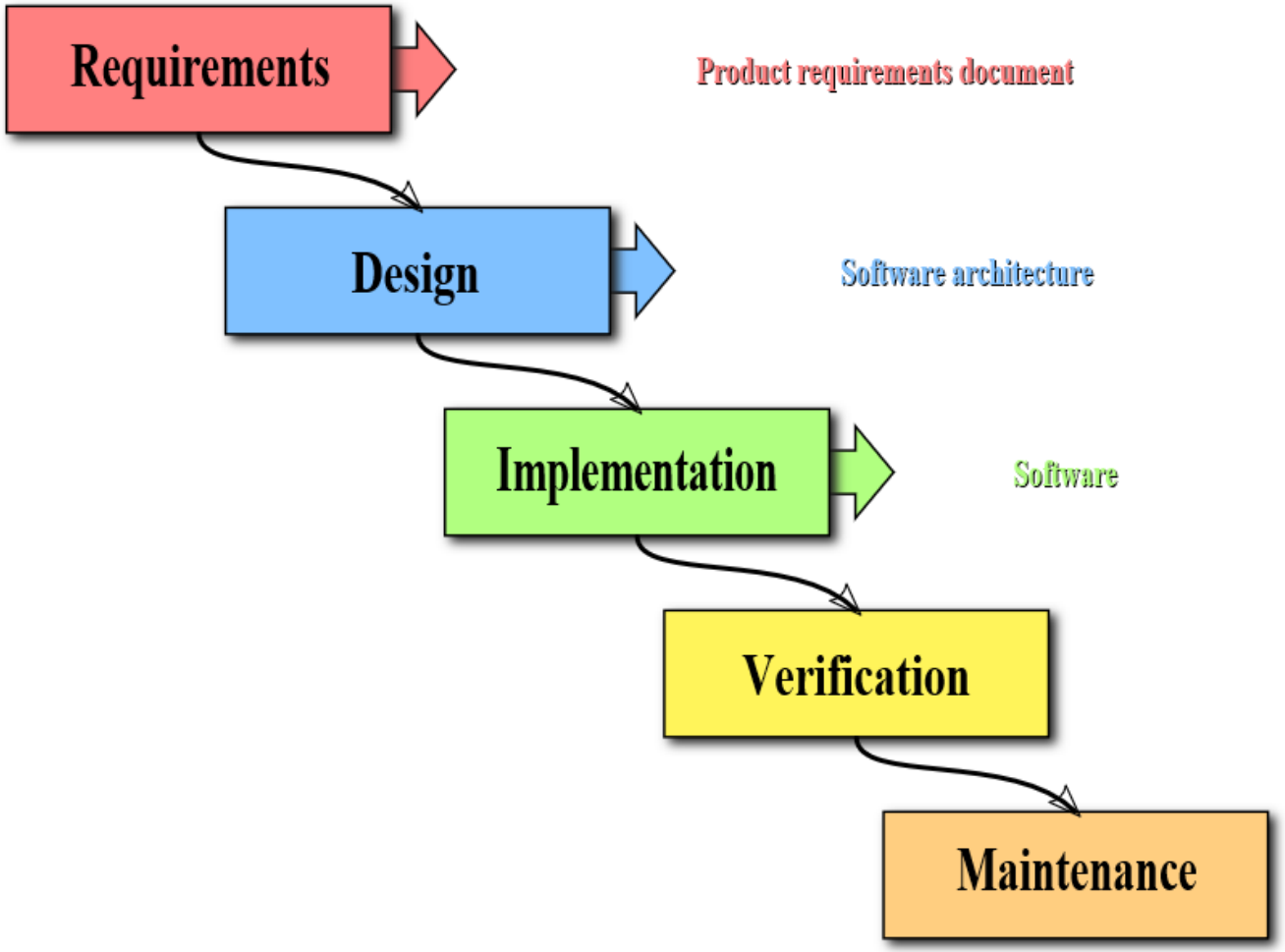
It is a simplified representation of a software process. We will talk about three widely used generic process models-

1. The Waterfall Model
2. The Evolutionary Model
3. The Component-based Model

The Waterfall Model- This model is a linear and sequential approach of software development where software develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase.

The fundamental development activities of this model are listed below. There is no overlapping of the phases.

1. Requirement Specifications.
2. Analysis and System Design.
3. Implementation and Unit Testing.
4. Integration and System Testing.
5. Operation and Maintenance.



The Waterfall Model

This mode is suitable for projects where-

- Requirements are clearly determined.
- Each phase is clearly laid out and sequentially flows to next phase.
- Each phase fully completes and makes available its output as input of next phase.

Advantages of Waterfall Model-

- **Departmentalization**- It divides the whole process into departments. This allows for separate schedules and deadlines for each department.
- It is easy to understand model.
- It is easy to manage model.
- Not a complex model.

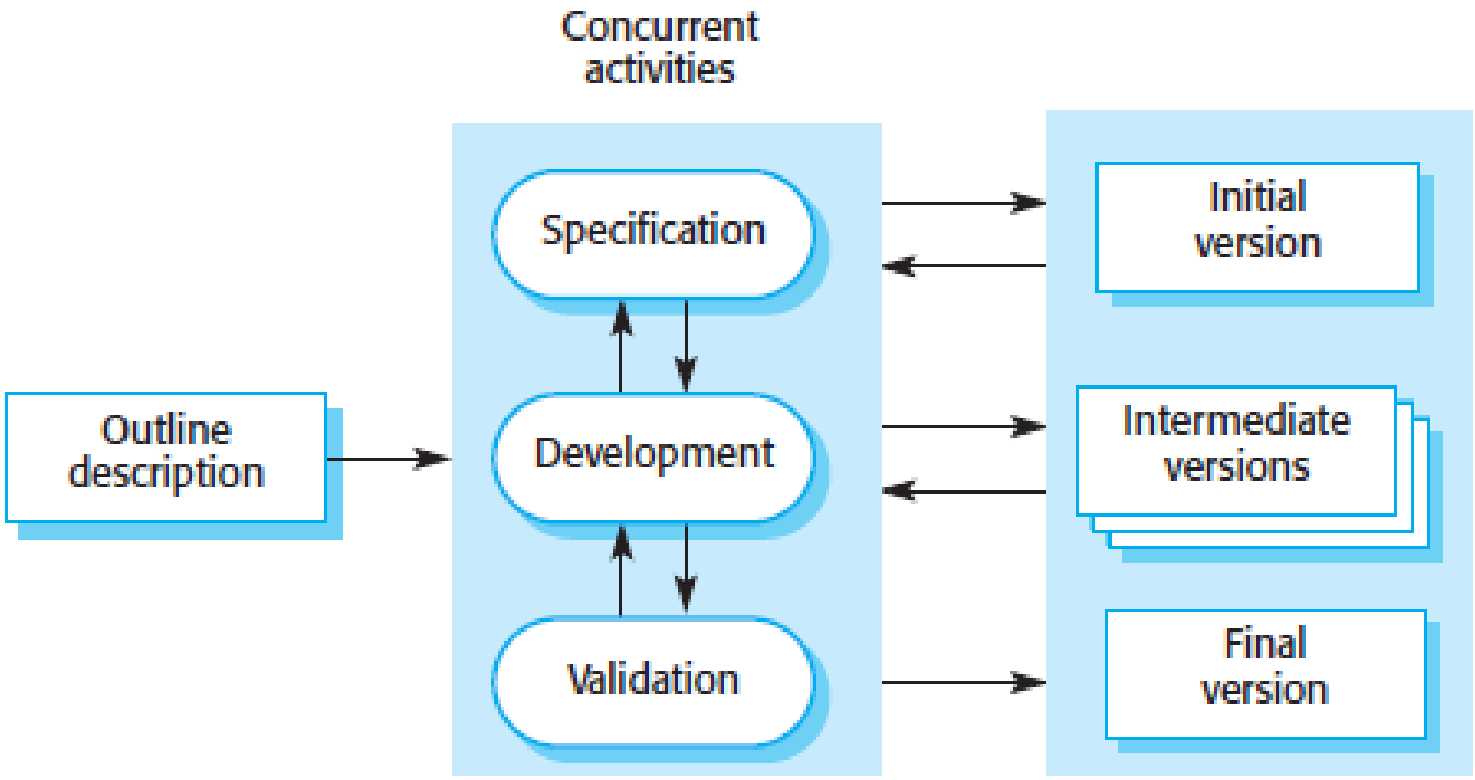
Disadvantages of Waterfall Model-

- No estimation of time and cost.
- Difficult to incorporate changes.
- Not for complex systems.

The Evolutionary Model-

This is a rapid software development model where an initial software implementation is rapidly developed from very abstract specifications, which is then iteratively modified according to the users' appraisal of the software.

In this software process, an initial software implementation is developed and given or shown to users. User works with this software implementation and gives comments and feedback based on which the system is refined and changes are incorporated to it. This process is repeated until a complete, full-fleged system is developed. Software specification, development and validation and testing activities are interleaved wih rapid feedback across activities.



The Evolutionary Model

Advantages of Evolutionary Model-

- It is useful to demonstrate the concept to prospective investors before actually developing the full system.
- It reduces risk of failure, as potential risks can be identified early by continuous appraisal of the software implementation and corrective steps can be taken.
- The development takes place with combined contribution of the development team and the client.
- User gets a fair idea of final product's working as a working model of the system is provided.
- User feedback is available at an early stage leading to better solution.

Disadvantages of Evolutionary Model-

- If changes suggested too may then it may disturb the rhythm of the development team.
- If the user demands too may changes then it may increase the complexity of the system.
- Repeated changes may increase the cost of development and it may derail the budgeting of the software system.

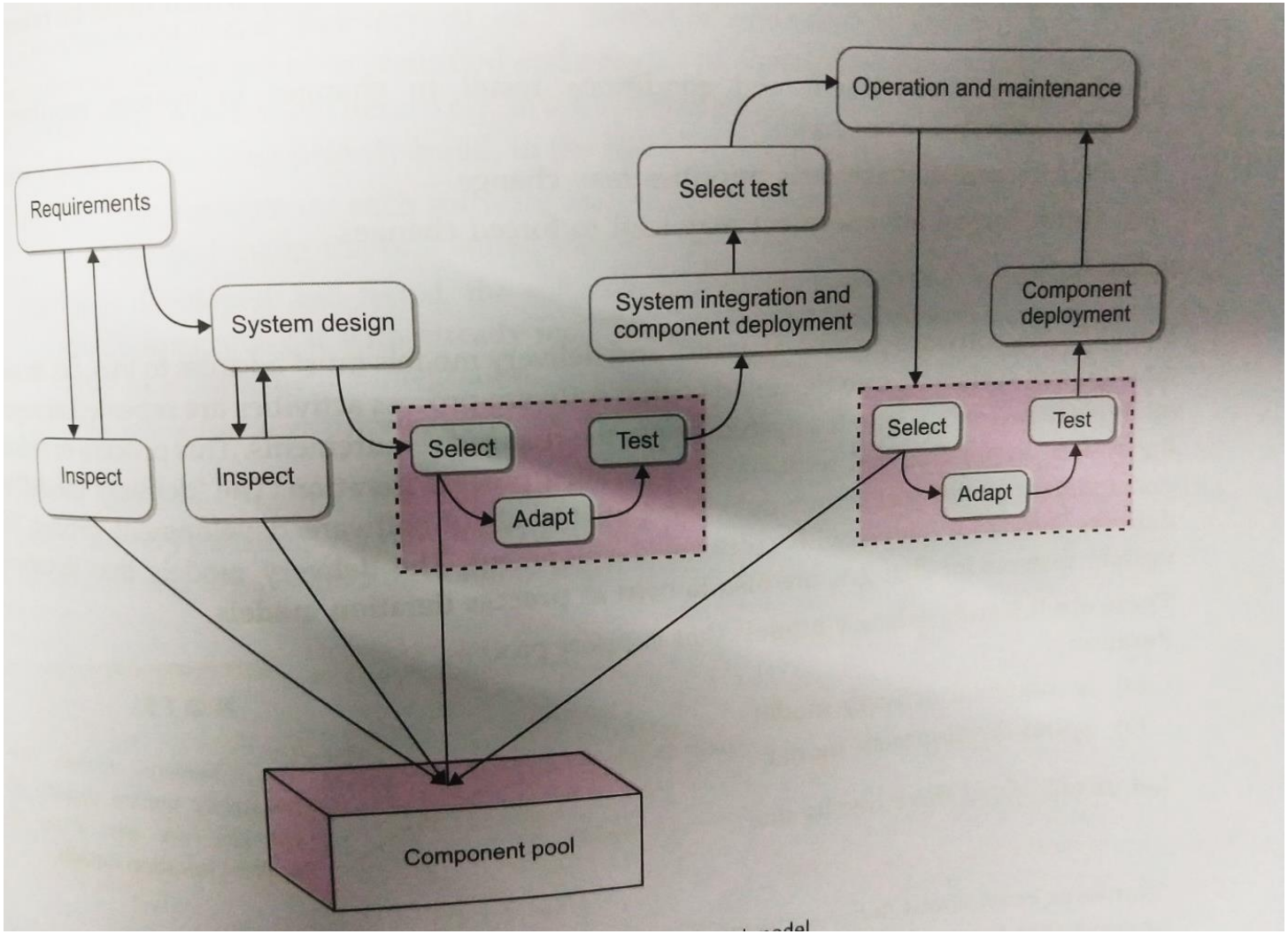
Component-Based Model-

It is a software element that confirms to a software model and can be independently deployed and composed without modification according to a composition standard.

While developing software, you may realize that a certain functionality is similar to a software component you developed earlier. This model is based on this very idea i.e., to incorporate and reuse existing software components in current software development if feasible and possible.

This model has following phases-

1. Component analysis.
2. Requirements modification.
3. System design with reuse.
4. Development and integration.



The Component-Based Model

Advantages of Component-based model-

- It reduces the amount of software to be developed.
- It results in reduced costs and risks, if reusable components are available.
- It also leads to faster delivery of the software.

Disadvantages of Component-based model-

- Sometimes requirements are compromised.
- This may lead to a system that does not meet the real needs of users.

Delivery Models-

Modern age software systems' delivery can never be once-and-for-all final delivery. Changes do occur and are unavoidable. Thus, software systems' delivery must incorporate the changes. So, modern day delivery systems are so designed so that software process activities are repeated at regular intervals to rework or redo the system as per the changed requirements. This process of redoing the system as per changed requirements is called process iteration.

The delivery models that iteratively update systems for changes , are also known as process iteration models.

There are two such delivery models that support process iteration-

a. Incremental delivery model.
b. Spiral development model.

Incremental delivery model-

This model is a development and delivery model that combines the strengths of two software process models- Waterfall Model and Evolutionary Model.

This model has mainly these work-phases-

1. **Determine overall services to be delivered.**
2. **Determine software incremental builds-** The following mechanism is carried out to determine the software increments or software builds, which broadly means the set of services to be delivered in one delivery-phase.

i. The customer is asked to priorities these services as the most important service(s) to the least important service(s).

ii. As per the priority levels, the number of software builds are defined where each software increment provides each priority level service(s). The allocation of services to increments depends on the priority of service.

3. **Development and delivery of each software incremental build- O**nce the system incremental builds have been identified and prioritized, then following activities are carried out for each software increment iteratively, in the order of their priority.

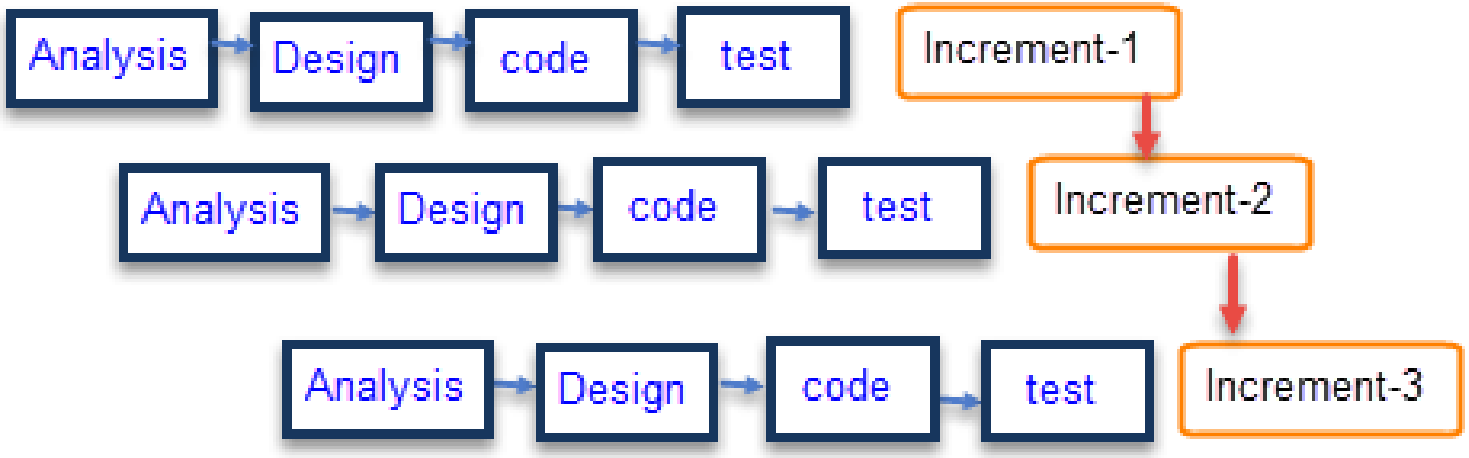
i. The requirements for each software increment are defined in details, and that increment is devloped using the best suited software process.

ii. Once developed and tested, the software build is delivered and implemented for the client. The system working is retested after new increment's integration.

iii. The work for next priority software build starts and whole process is repeated.

Advantages:-

1. It generates working software quickly.
2. It is more flexible, it costs less to change scope and requirements.
3. It is easier to test and debug during a smaller iteration.
4. There is a lower risk of overall project faliure as tested increments are added.
5. Easier to manage risk because risky pieces are identified and handeled first of all.



The Incremental delivery Model

Disadvantages-

1. After each software increment is added with the system, the integration testing to test the system working as a whole, is carried out, which increases the testing load.
2. Every software increment is developed seprately. Each such phase is rigid and do not overlap each other.
3. If all requirements are not clearly identified and not aptly prioritised , it may lead to major problems in overall system architecture.

The Spiral Model-

In this model, the software is developed in a series of incremental releases with the early stages being either paper models or prototypes.

The sequence of activities that takes place in this model, takes place with some backtracking from one activity to another, just like a spiral- and hence the name.

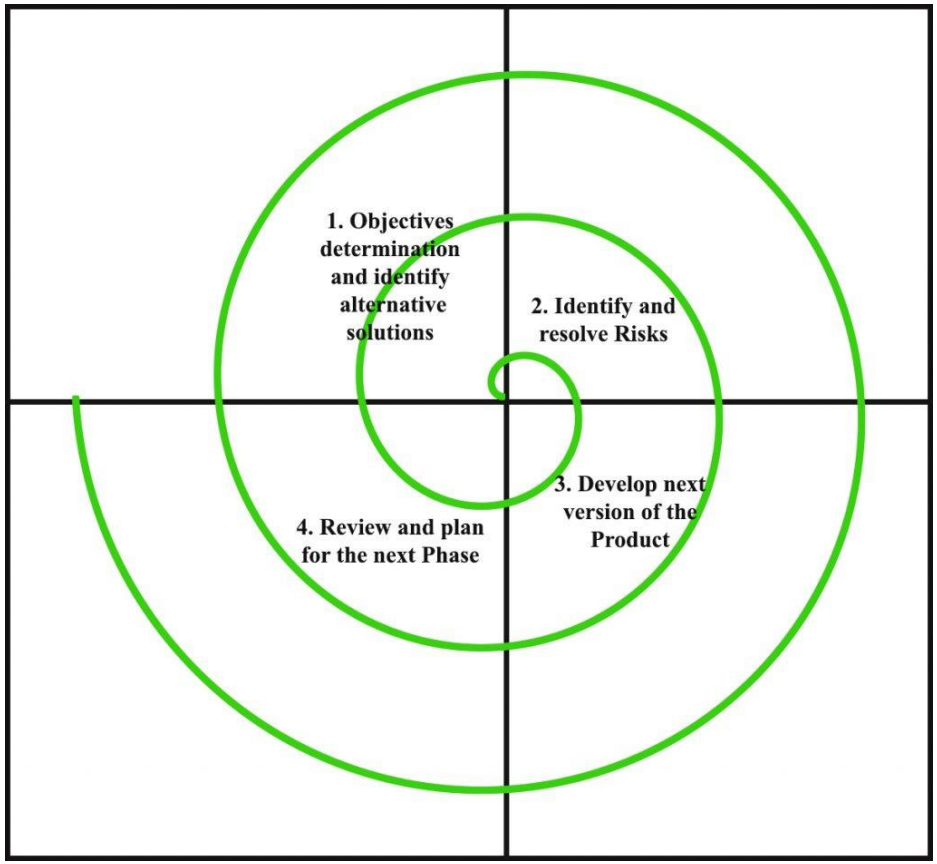
The sequence of activities of this model are represented through a spiral, where each loop corresponds to a phase of the software process. Thus, the innermost loop might be concerned with system feasiblity, the next loop with requirements definitions, the next loop with system design and so on.

The following activities are carried out during each phase of a spiral model-

1. **First Quadrant(Objective setting)**- in this phase, the objectives of the phases are determined and associated risks are examined.
2. **Second Quadrant(Risk assessment and reduction)**- in this phase, a detailed analysis is carried out for each identified risk. This phase is also responsible for risk reduction, hence measures are taken for risk reduction wherevever possible.

3. **Third Quadrant(Development and Validation)-** This phase is for thedevelopment and validation of the next level of the product after resolving the identified risks.

4. **Fourth Quadrant (Review and Planning)-** During this phase, the results achieved so far are reviews with the customer. And the planning for the next iteration around the spiral also takes place.



The Spiral Model

The spiral model is advantageous for the developement of large, complex, and expensive software systems.

It is also suitable to systems where reaction to risks at each evolutionary level is required.

Thank you

Please follow us on our blog

www.pythontrends.wordpress.com