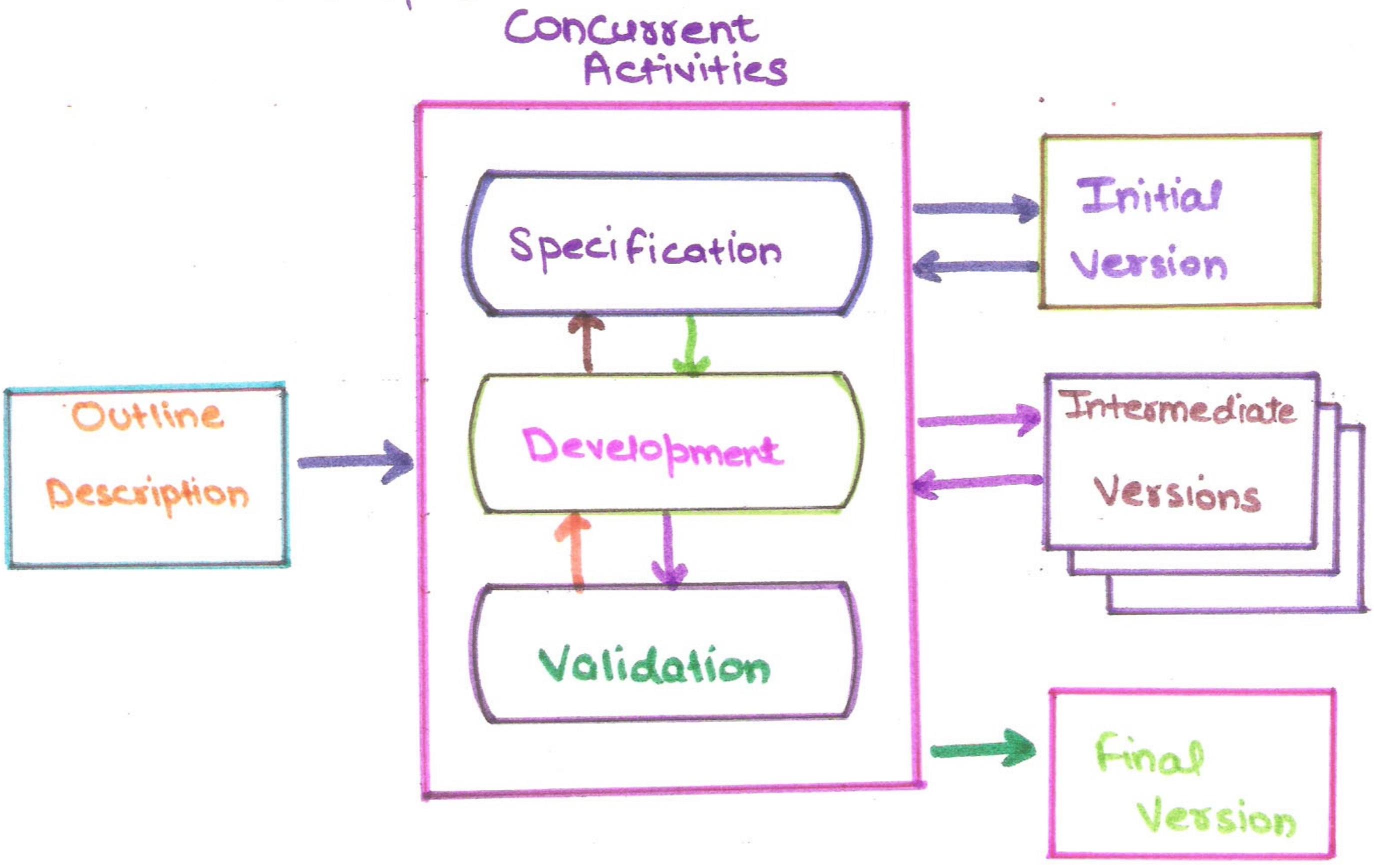
# Software Process Model

## Evolutionary Model

Evolutionary Model is based on the initial implementation will result in the user Comments, So it can be Repaired through many Versions until an adequate System Can be developed.

In addition to having separate activities, this model provides feedback to Developers.



→ Evolution Process Model

Computer Science Lectures By ER·Deepak Garg

The Evolutionary model **DIVIDES** the development cycle into **SMALLER**, Incremental waterfall Models in which users are able to get access to the product at the end of each cycle.

The users provide feedback on the product for **Planning stage** of the next cycle and the development team responds, often by changing the product, plans, or process.

These Incremental cycles are typically **Two to four weeks** in duration and continue until the product is shipped.

- These models are more suited to object oriented systems.
- They are iterative.
- They enable the software developer to develop increasingly more complex versions of the software.
- Like all complex systems, software evolve over a period of time and hence evolutionary models are more suited to software development.
- Requirements change while software gets developed.

# Advantages

**Errors Reduction:-** As the version is tested with Customer which reduces the Error thoroughly.

**User Satisfaction :-** User gets satisfy as he gets the full chance of experimenting partially developed System.

**Bussiness Benefits :** Successful use of this model Can benefit not only Business Results but Marketing and internal operations as well.

**High Quality :-** As user gets satisfy with every version, it produces the high Quality product.

**Low Risk !-** There is significant Reduction of Risk as versions is implemented. This Risks may be associated with

- Missing Schedule Deadline
- wrong Feature sets
- Poor Quality

**Reduction in Cost:-** Some Design issues are cheaper to Resolve through Experimentation than through Analysis. It Reduces Cost by providing Structured and Discipline Avenue for Experimentation.

# Disadvantages

**Several Version Release :-** Developers has to make Several versions which increases their **Efforts.**

**Dividing Software :-** It is difficult to Divide the Software and problems in several versions that would be acceptable to the Customer which can be implemented and delivered incrementally.

**Uncertain Nature of Customer Needs:** A Confused user has uncertainity over his requirements, So giving Several Versions may Change his Requirement Rapidly.

**Time and Cost :-** As this model Reduces **Time and cost** but Requirement is not gathered Correctly, it will Subsequently increases the Time , cost and Efforts.

**Confusion by Several version :-** An user might get **Confused** by Several **Versions of Software.** It will effect on the **final product**.