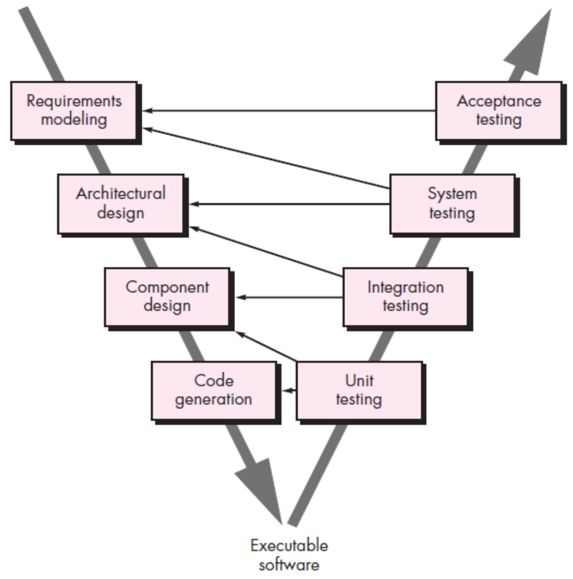
V-Model

V model is a variant of waterfall model. Just like waterfall model, name is in accordance with its shape.
V is for **Verification** (continuous process of checking) & **Validation** (done only once after product is generated).



- As software team moves down the left side of the V, basic problem requirements are refined into progressively more detailed and technical representations of the problem and its solution.
- Test cases are designed in each development phase.
- Once code has been generated, the team moves up the right side of the V, essentially performing a series of tests (quality assurance actions) that validate each of the models created as the team moves down the left side.

1. **Requirement modeling-Acceptance Test**
   - **Requirements modeling** identifies the requirements that a software application or system must meet in order to solve the business problem.
     Along with requirement gathering and SRS preparation, Acceptance test plan/design is made.
   - **User Acceptance Test** (UAT) is performed in a user environment that resembles the production environment, using realistic data.
     UAT verifies that delivered system meets user's requirement and system is ready for use in real time.

2. **Architectural Design-System Test**
   - **Architectural design:** The architecture of a software system is analogous to the architecture of a building.
     It is a blueprint for the system to be developed.
     The tasks to be executed by design team are decided.
     And System Test design is made.
   - **System Test** ensures that expectations from application developed are met.
     The whole application is tested for its functionality, interdependency and communication.
     System Testing verifies that functional and non-functional requirements have been met.

3. **Component Design-Integration Test**
    - **Component Design**: Component design include the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology details etc.
      The integration testing design is carried out.
    - **Integration tests** are performed to test the coexistence and communication of the internal modules within the system.

4. **Code generation-Unit Test**
    - **Code generation**: The actual coding of the system modules designed in the previous phase is taken up in the Coding phase.
      The best suitable programming language is decided based on the system and architectural requirements.
      The coding is performed based on the coding guidelines and standards.
      Unit Test plan is also made.
    - **Unit testing** is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

Advantages of V-model over iterative waterfall

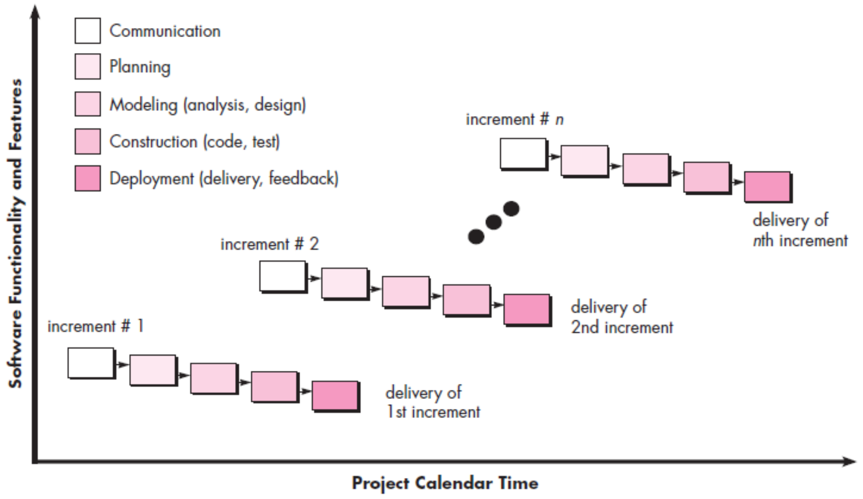
- **Shorter testing phase**
    - In the V-model, much of the testing activities (test case design, test planning, etc.) are carried out in parallel with the development activities.
    - Before testing phase starts significant part of the testing activities, including test case design and test planning, is already complete.
    - This model usually leads to a shorter testing phase and an overall faster product development as compared to the iterative model.

- **Efficient manpower utilization**
    - Since test cases are designed when the schedule pressure has not built up, the quality of the test cases are usually better.
    - The test team is reasonably kept occupied throughout the development cycle in contrast to the waterfall model where the testers are active only during the testing phase.
    - This leads to more efficient manpower utilization.

- Testing team has better understanding as they are associated with the project from the beginning which helps them to carry out **effective testing** of the software.

V-Model has all the **drawbacks** of Waterfall as it is derived from it.

Incremental Process Model


Communication
Planning
Modeling (analysis, design)
Construction (code, test)
Deployment (delivery, feedback)
increment # n
delivery of nth increment
increment # 2
delivery of 2nd increment
increment # 1
delivery of 1st increment
Software Functionality and Features
Project Calendar Time

- The incremental model combines elements of **linear and parallel process flows**.
- The incremental model applies **the waterfall (or iterative waterfall) model incrementally**.
- Each linear sequence produces deliverable "increments" of the software in a manner that is similar to the increments produced by an evolutionary process flow.

- The first increment is called **a core product** that tackles highest priority requirements. (core requirements)
- Many supplementary features (known or unknown) remain undelivered.
- The core product is used by the customer (or undergoes detailed evaluation).
- From the **feedback** from the customer, a plan is developed for the next increment.
- Each delivered version (increment) of the software incorporates **additional features** over the previous version (increment) and also **refines (modifies) the features** that were already delivered to the customer.
- This process is repeated after the delivery of each increment, until the complete product is produced.

It is important to note that an **incremental philosophy** is also used for all **"agile" process models.**

Why before the delivery of an increment, communication and planning activities of next increment are started? (As in the diagram)

- This is done to achieve better resource management, to minimize various risks and to give early delivery of the product to the customer.

Risks can be **project risk** (schedule slippage, resource issues, and customer issues), **technical risks** (design issue, implementation issues, and testing issues) and **business risks** (budgetary failures, unusable product development, etc)

Benefits of Incremental model:

1. It becomes easier to accommodate request for change from the customers.
    - At any time, planning is done only for the next increment and no long-term plans are made.
2. Errors are reduced.
    - Each increment delivered gets tested thoroughly as it is used by the customer.
    - Targeted and rigorous testing is possible as few changes are made within any single iteration.
    - This reduces the chances of errors in final product, making the software reliable.
3. Better resource management
    - For developing organization, it is difficult to deploy (arrange) large resources and manpower in one go. Incremental model facilitates gradual commitment of resources.
4. Customer gets important functionality early as initial product delivery is fast.
5. Initial delivery cost is less.

Drawbacks of Incremental model:

1. Requirements must be clear and predefined.
2. Resulting cost may not be low.
3. As additional functionality can be added, problems may arise related to system architecture which may not be known earlier.
4. It is applicable to a limited kind of software/systems.
    - Not suitable for system software (like operating system) and embedded software.

When to use Incremental Model over Waterfall model(s)?

- Most of the requirements are known up-front but are expected to evolve over time.
- The requirements are prioritized.
- System can be separated in parts.
- There is a need to get the basic functionality delivered fast.
- A project has lengthy development schedules.
- A project has new technology.
- The domain is new to the team.

Example of a system that uses incremental model:

- Microsoft Office provides various products like Word, PowerPoint, Excel, etc.
- Every version of **MS Word** is not same and doesn’t have same features.
- Engineers will keep on updating features on the software for every release to make productivity.
- Engineers uses incremental model to develop MS Word, because each and every version of MS Word changes based on adding or removing features.