

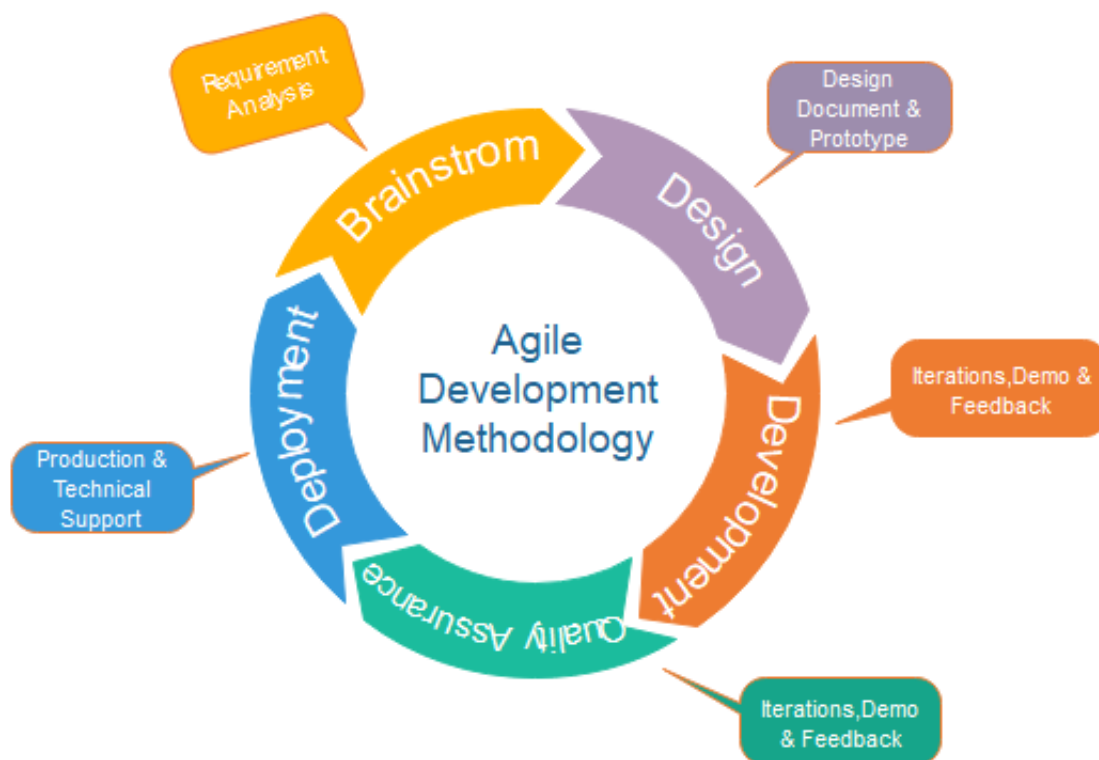
## **Agile Model : -**

### **Introduction : -**

The meaning of Agile is swift or versatile. "**Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

### **Diagram :-**



***Fig. Agile Model***

### **Advantages: -**

- 1) Frequent Delivery
- 2) Face-to-Face Communication with clients.
- 3) Efficient design and fulfils the business requirement.
- 4) Anytime changes are acceptable.
- 5) It reduces total development time.

### **Disadvantages: -**

- 1) Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
- 2) Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

### **Real life Example:-**

- 1) Restaurant orders:

Preparation of some of the food before opening the shop (sprint planning)

continuous delivery of orders (ad hoc stories)

number of successful orders (velocity)

- 2) cricket team:

Run rate (velocity)

team (scrum team self sufficient)

over (sprint length)

captain/ coach (scrum master)

### **'V' shaped model :-**

#### **Introduction :-**

V-Model also referred to as the Verification and Validation Model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.

**Verification:** It involves a static analysis method (review) done without executing code. It is the process of evaluation of the product development process to find whether specified requirements meet.

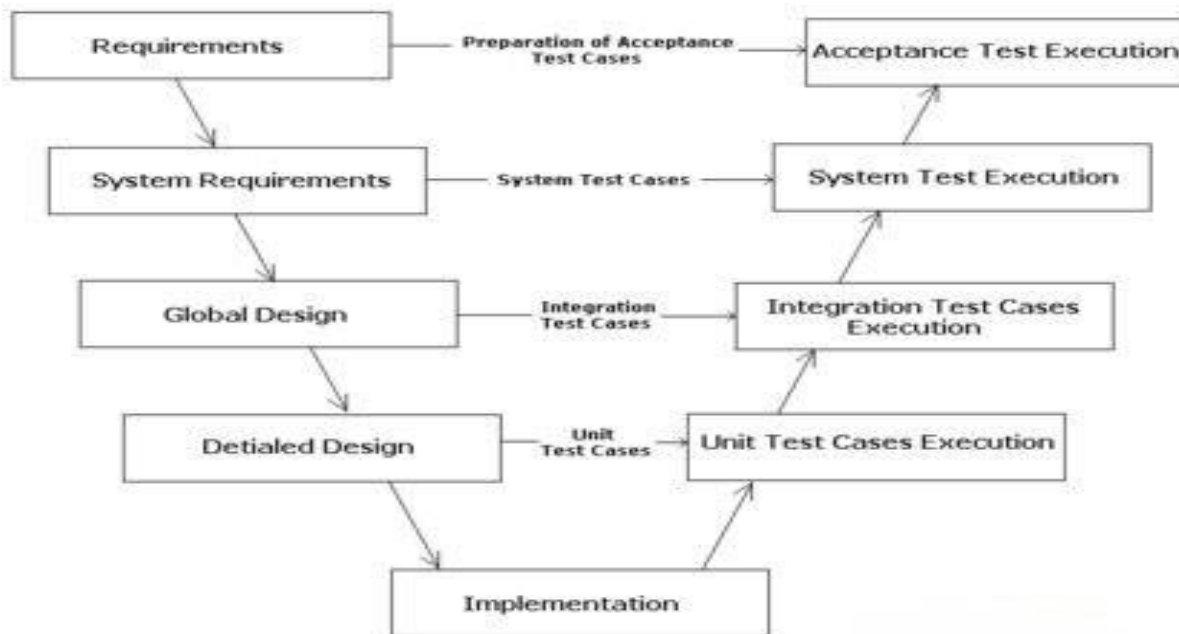
**Validation:** It involves dynamic analysis method (functional, non-functional), testing is done by executing code. Validation is the process to classify the software after the completion of the development process to determine whether the software meets the customer expectations and requirements.

So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation process is joined by coding phase in V-shape. Thus it is known as V-Model.

#### **Purpose of V shaped Model :-**

- 1) When the requirement is well defined and not ambiguous.
- 2) The V-shaped model should be used for small to medium-sized projects where requirements are clearly defined and fixed.
- 3) The V-shaped model should be chosen when sample technical resources are available with essential technical expertise.

### Diagram (V-Model ) : -



### Advantages:-

- 1) This is the highly-disciplined model and phases are completed one at a time
- 2) Works well for smaller projects where requirements are very well understood.
- 3) Simple and easy to understand and use.
- 4) Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

### Disadvantages

- 1) High risk and uncertainty.
- 2) Not a good model for complex and object oriented projects.
- 3) Poor model for long and ongoing projects.
- 4) Not suitable for the projects where requirements are at a moderate to high risk of changing.
- 5) No working software is produced until late during the life cycle.

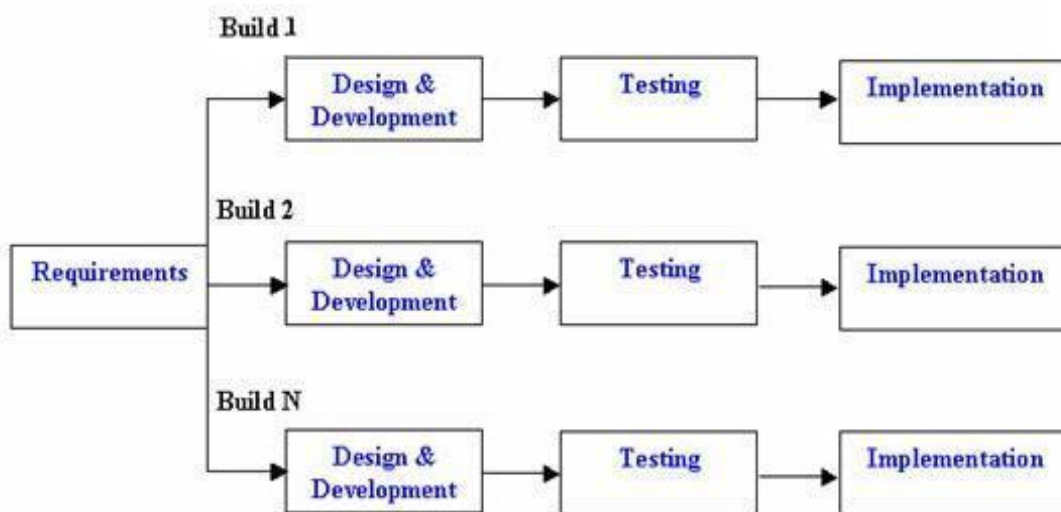
### Real-life example:-

- 1) Projects initiated from a request for proposals (RFPs), the customer has a very clear documented requirements
- 2) Military projects
- 3) Mission Critical projects, for example, in a Space shuttle
- 4) Embedded systems.

## Incremental Model :-

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Incremental model is a type of software development model like V-model, Agile model etc.

In this model, each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.



### Purpose:

- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- A new technology is being used
- Resources with needed skill set are not available
- There are some high risk features and goals.

### Advantages :

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

### Disadvantages of Incremental model:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Total cost is higher than waterfall.

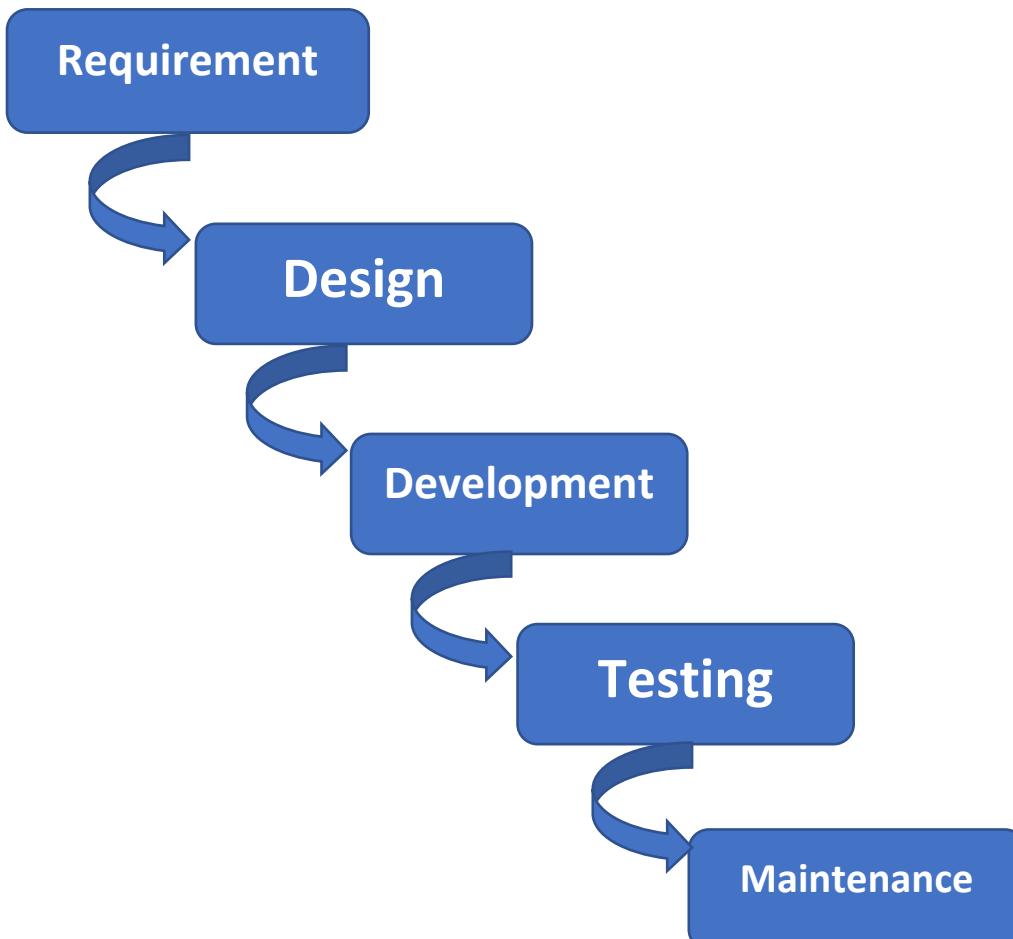
### Real Time Example:

- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- A new technology is being used.
- Resources with needed skill set are not available.
- There are some high-risk features and goals.

## Waterfall model: →

Winston Royce introduced the Waterfall Model.

Always follow the step in this order and do not overlap. This model consider that one phase can started after the completion of the previous stage. This model is named "**Waterfall Model**", because its diagrammatic representation resembles a cascade of waterfalls.



## Five phases of model:

### 1. Requirements:

During this phase, detailed requirements of the software system to be developed are gathered from client

### 2. Design Phase:

Plan the programming language, for Example [Java](#), [PHP](#), .net or database like Oracle, MySQL, etc. or other high-level technical details of the project

### 3. Development:

After design stage, it is built stage, that is nothing but coding the software

### 4. Testing:

In this phase, you test the software to verify that it is built as per the specifications given by the client.

### 5. Maintenance:

Once your system is ready to use, you may later require change the code as per customer request

### where the use of the Waterfall model is most suited are:

- When the requirements are constant and not changed regularly.
- A project is short
- The situation is calm
- Where the tools and technology used is consistent and is not changing
- When resources are well prepared and are available to use.

### Advantages:

- Before the next phase of development, each phase must be completed
- Any changes in software is made during the process of the development
- They should perform quality assurance test (Verification and Validation) before completing each stage
- Suited for smaller projects where requirements are well defined

### Disadvantages:

- Error can be fixed only during the phase
- It is not desirable for complex project where requirement changes frequently

- Small changes or errors that arise in the completed software may cause a lot of problems
- Documentation occupies a lot of time of developers and testers

### Example:

In the olden days, Waterfall model was used to develop enterprise applications like Customer Relationship Management (CRM) systems, Human Resource Management Systems (HRMS), Supply Chain Management Systems, Inventory Management Systems, Point of Sales (POS) systems for Retail chains etc.

Value chain of CVS in reasturant industries, point of Sales (POS) systems for Retail chains:

Headquarter -> Order to distributor -> Delivery -> CVS store -> Display -> Purchase.

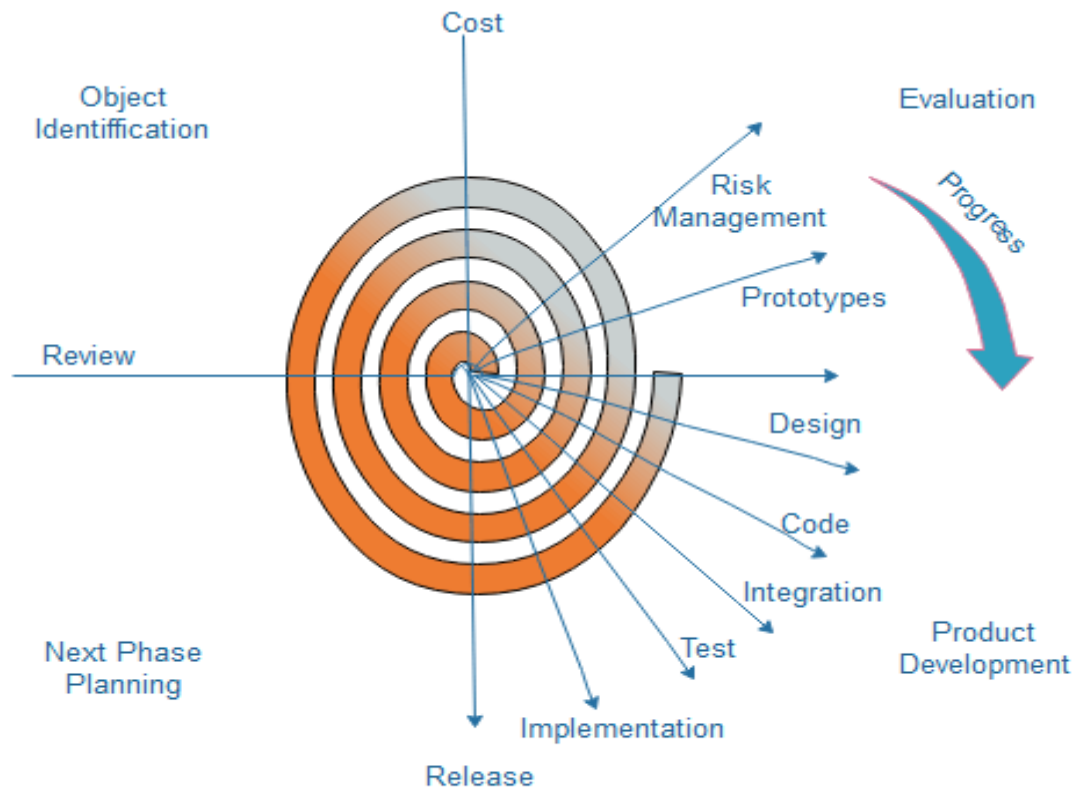
## Spiral Model

- The spiral model, initially proposed by Boehm
- The spiral model is used for large, expensive and complicated projects.
- It implements the potential for rapid development of new versions of the software.

### Purpose of model :-

- projects in which changes may be required at any time.
- long term projects that are not feasible due to altered economic priorities.
- medium to high risk projects.
- projects in which cost and risk analysis is important.
- projects that would benefit from the creation of a prototype .
- projects with unclear or complex requirements.

## Diagram :-



**Fig. Spiral Model**

- **Objective setting:** Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.
- **Risk Assessment and reduction:** The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.
- **Planning:** Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.
- The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.
- The risk-driven feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach. The spiral model works for development as well as enhancement projects.

State when we use

- When deliverance is required to be frequent.
- When the project is large
- When requirements are unclear and complex
- When changes may require at any time
- Large and high budget projects



## Advantages

- High amount of risk analysis
- Useful for large and mission-critical projects.

## Disadvantages

- Can be a costly model to use.
- Risk analysis needed highly particular expertise
- Doesn't work well for smaller projects.

## Real time examples :-

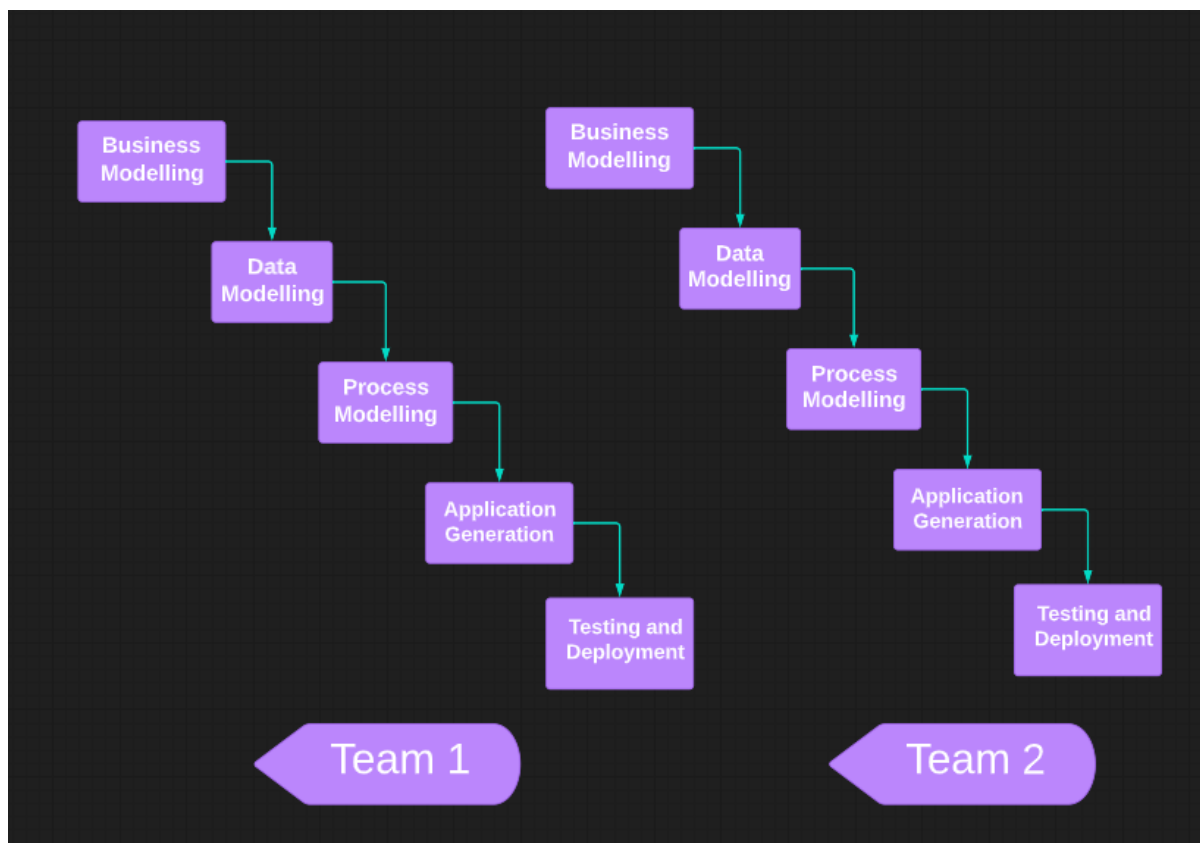
- Microsoft used it to develop early versions of Windows.
- The Gantt chart software was also made using spiral model.
- Game development is another industry who uses spiral model to develop the games.

## Rapid application development (RAD)

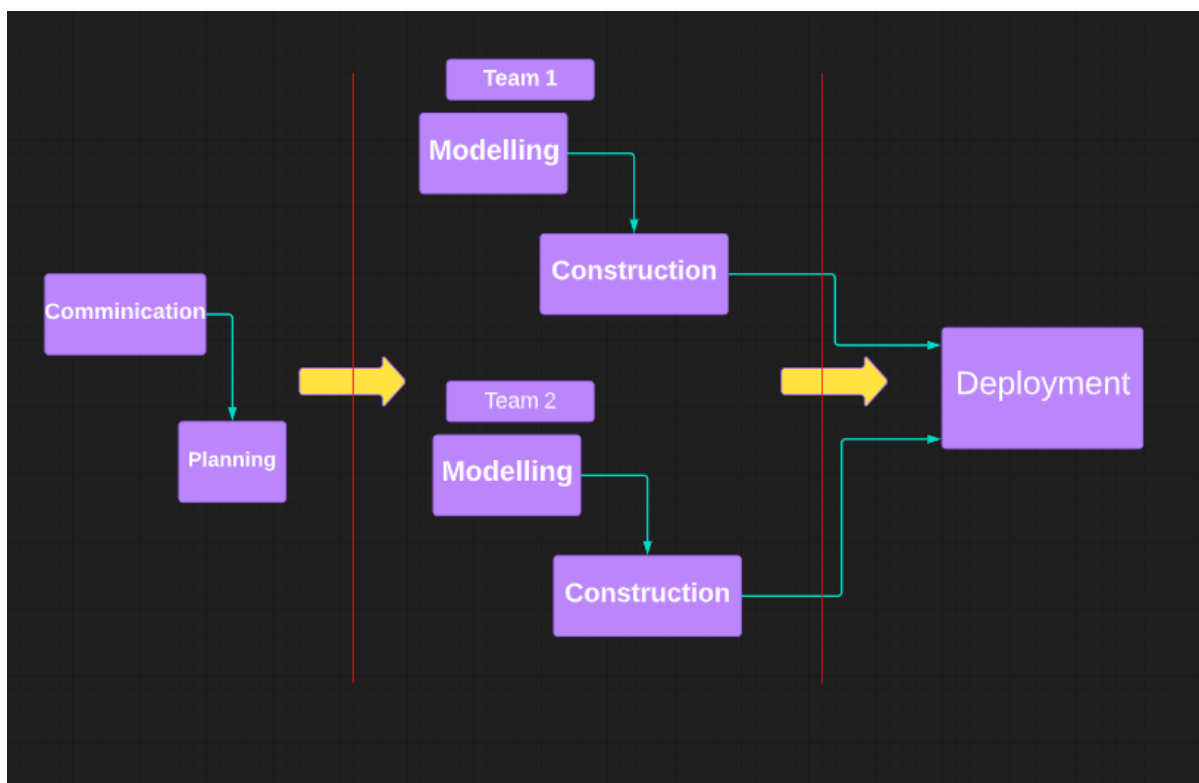
RAD basically emphasizes on short development cycle (60-90 days)

### When to use RAD Model?

When the customer has well-known requirements, the user is involved throughout the life cycle, the project can be **time bounded**, the functionality delivered in increments, high performance is not required, low technical risks are involved and the system can be modularized. In these cases, we can use the RAD Model.



1. **Requirements:** Planning The information flow among business functions is defined by answering questions like what data drives the business process, what data is generated, who generates it, where does the information go, who process it and so on.
2. **Data Modelling:** The data collected from business modeling is refined into a set of data objects (entities) that are needed to support the business. The attributes (character of each entity) are identified, and the relation between these data objects (entities) is defined.
3. **Process Modelling:** The information object defined in the data modeling phase are transformed to achieve the data flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting, or retrieving a data object.
4. **Application Generation:** Automated tools are used to facilitate construction of the software; even they use the 4th GL techniques.
5. **Testing & Turnover:** Many of the programming components have already been tested since RAD emphasis reuse. This reduces the overall testing time. But the new part must be tested, and all interfaces must be fully exercised.



### Advantages:

- The use of reusable components helps to reduce the cycle time of the project.
- Feedback from the customer is available at the initial stages.
- Reduced costs as fewer developers are required.
- The use of powerful development tools results in better quality products in comparatively shorter time spans.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.

### Disadvantages:

- The use of powerful and efficient tools requires highly skilled professionals.
- The absence of reusable components can lead to the failure of the project.
- The team leader must work closely with the developers and customers to close the project in time.
- The systems which cannot be modularized suitably cannot use this model.
- Customer involvement is required throughout the life cycle.
- It is not meant for small-scale projects as in such cases, the cost of using automated tools and techniques may exceed the entire budget of the project.

### Applications:

1. This model should be used for a system with known requirements and requiring a short development time.
2. It is also suitable for projects where requirements can be modularized and reusable components are also available for development.
3. The model can also be used when already existing system components can be used in developing a new system with minimum changes.
4. This model can only be used if the teams consist of domain experts. This is because relevant knowledge and the ability to use powerful techniques are a necessity.
5. The model should be chosen when the budget permits the use of automated tools and techniques required.

## Comparison between the Models

Properties of Model	Water-Fall Model	Incremental Model	Spiral Model	RAD Model	Agile Model	V shaped Model
Planning in early stage	Yes	Yes	Yes	No	No	Yes
Returning to an earlier phase	No	Yes	Yes	Yes	Yes	No
Handle Large-Project	Not Appropriate	Not Appropriate	Appropriate	Not Appropriate	Yes	Not appropriate
Detailed Documentation	Necessary	Yes, but not much	Yes	Limited	Yes	Yes
Cost	Low	Low	Expensive	Low	Low	Expensive
Requirement Specifications	Beginning	Beginning	Beginning	Time boxed release	Time boxed release	Beginning
Flexibility to change	Difficult	Easy	Easy	Easy	Easy	Difficult
User Involvement	Only at beginning	Intermediate	High	Only at the beginning	High	Only at the beginning
Maintenance	Least	Promotes Maintainability	Typical	Easily Maintained	Easily Maintained	Easily maintained
Duration	Long	Very long	Long	Short	Depends on project	Long
Risk Involvement	High	Low	Medium to high risk	Low	Low	Moderate to high
Framework Type	Linear	Linear + Iterative	Linear + Iterative	Linear	Iterative and incremental	Sequential
Testing	After completion of coding phase	After every iteration	At the end of the engineering phase	After completion of coding	After every iteration	After every iteration
Overlapping Phases	No	Yes (As parallel development is there)	No	Yes	No	No
Re-usability	Least possible	To some extent	To some extent	Yes	Yes	No
Time-Frame	Very Long	Long	Long	Short	Long	Ideal time
Working software availability	At the end of the life-cycle	At the end of every iteration	At the end of every iteration	At the end of the life cycle	At the end of every iteration	At the end of every iteration
Team size	Large Team	Not Large Team	Large Team	Small Team	Large team	Large team

