DevOps

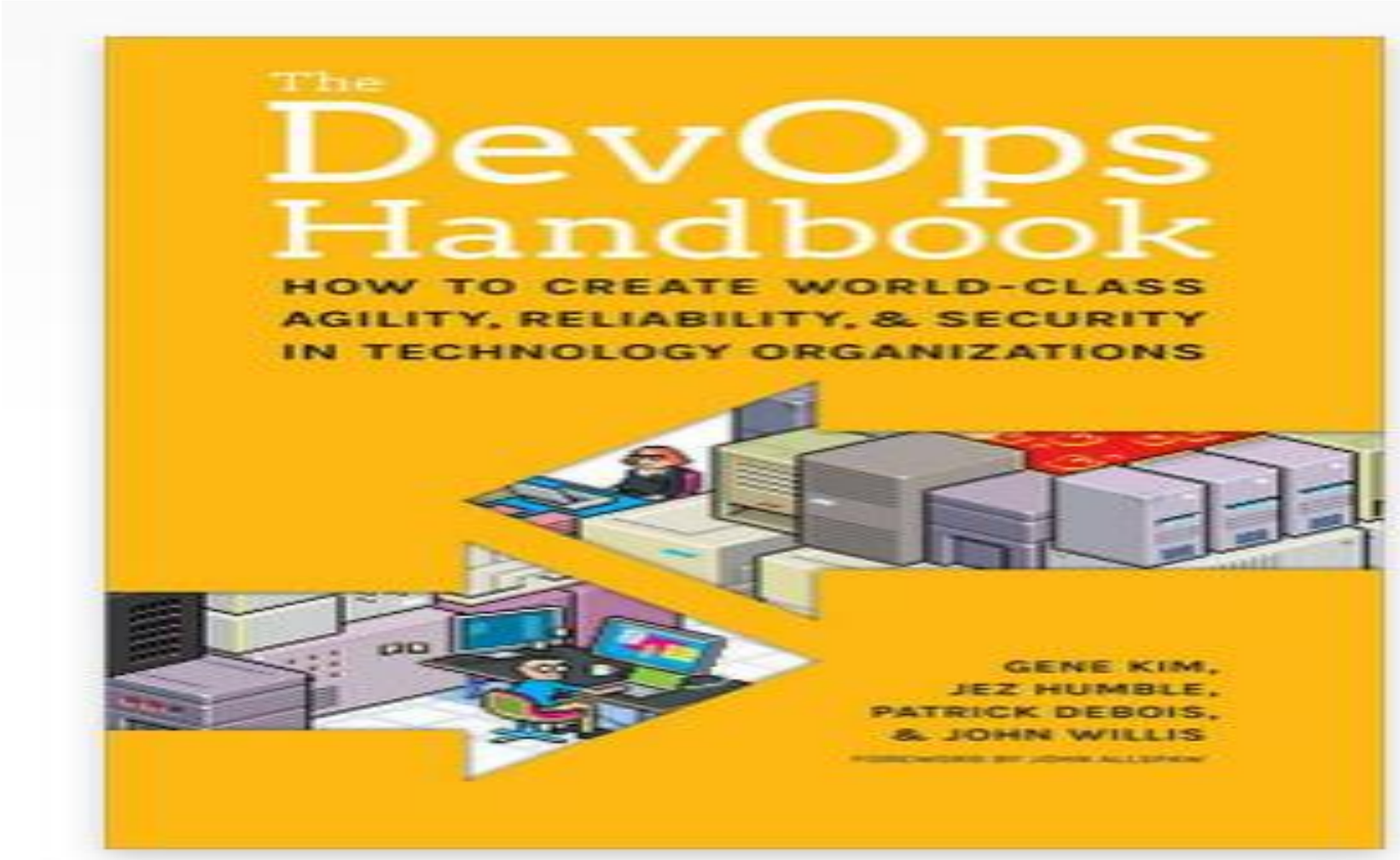
(Unit No -1)

Topics : Phases of Software Development life cycle. Values and principles of agile software development.

Dr. K. Sandeep
Assoc. Prof & Head
Department of I.T

DevOps

Text Books



Software Development Life Cycle (SDLC)

- A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle.

- A life cycle model represents all the methods required to make a software product transit through its life cycle stages. It also captures the structure in which these methods are to be undertaken.

- In other words, a life cycle model maps the various activities performed on a software product from its inception to retirement.

Need of SDLC

- The development team must determine a suitable life cycle model for a particular plan and then observe to it.
- Without using an exact life cycle model, the development of a software product would not be in a systematic and disciplined manner.
- When a team is developing a software product, there must be a clear understanding among team representative about when and what to do. Otherwise, it would point to chaos and project failure.
- This problem can be defined by using an example. Suppose a software development issue is divided into various parts and the parts are assigned to the team members.
- From then on, suppose the team representative is allowed the freedom to develop the roles assigned to them in whatever way they like.

- It is possible that one representative **might start writing the code** for his part, another might choose to prepare the **test documents first**, and some other engineer **might begin with the design phase** of the roles assigned to him. This would be one of the perfect methods for **project failure**.
- Without software life cycle models, it becomes tough for software project managers to monitor the progress of the project.
- **SDLC Cycle** represents the process of developing software.
- SDLC framework includes the following steps:

The stages of SDLC are as follows:

Stage1: Planning and requirement analysis

Stage2: Defining Requirements

Stage3: Designing the Software

Stage4: Developing the project

Stage5: Testing

Stage6: Deployment

Stage7: Maintenance

Stage1: Planning and requirement analysis

- Requirement Analysis is the most important and necessary stage in SDLC.
- In this phase, **all the requirements are collected from the customer/client**. They are provided in a document called **Businessmen requirement specification (BRS)** and **System requirement specification (SRS)**. All the details are discussed with the customer/client in detail.
- The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry. Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.
- Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

Stage1: Planning and requirement analysis (Cont..)

- **For Example**, A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.
- Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.
- Once the requirement is understood, the **SRS (Software Requirement Specification)** document is created. The developers should thoroughly follow this document and should be reviewed by the customer for future reference.

Stage2: Defining Requirements

- Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.
- This is accomplished through "**SRS"- Software Requirement Specification document** which contains all the product requirements to be constructed and developed during the project life cycle.
- **High-level design (HLD):** It gives the architecture of software products.
- **Low-level design (LLD):** It describes how each feature in the product should work and every component.

Stage3: Designing the Software

- The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.
- This is the longest phase.
- This phase consists of Front end + Middleware + Back-end
- **In front-end:** Development of coding is done even SEO settings are done.
- **In Middleware:** They connect both the front end and back end.
- **In the back-end:** A database is created.

Stage4: Developing the project

- In this phase of SDLC, **the actual development begins, and the programming is built**. The implementation of design begins concerning writing code. Developers must follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Stage5: Testing

- Testing is carried out to verify the entire system. The aim of the tester is to find out the gaps and defects within the system and also to check whether the system is running according to the requirement of the customer/client.

Stage6: Deployment

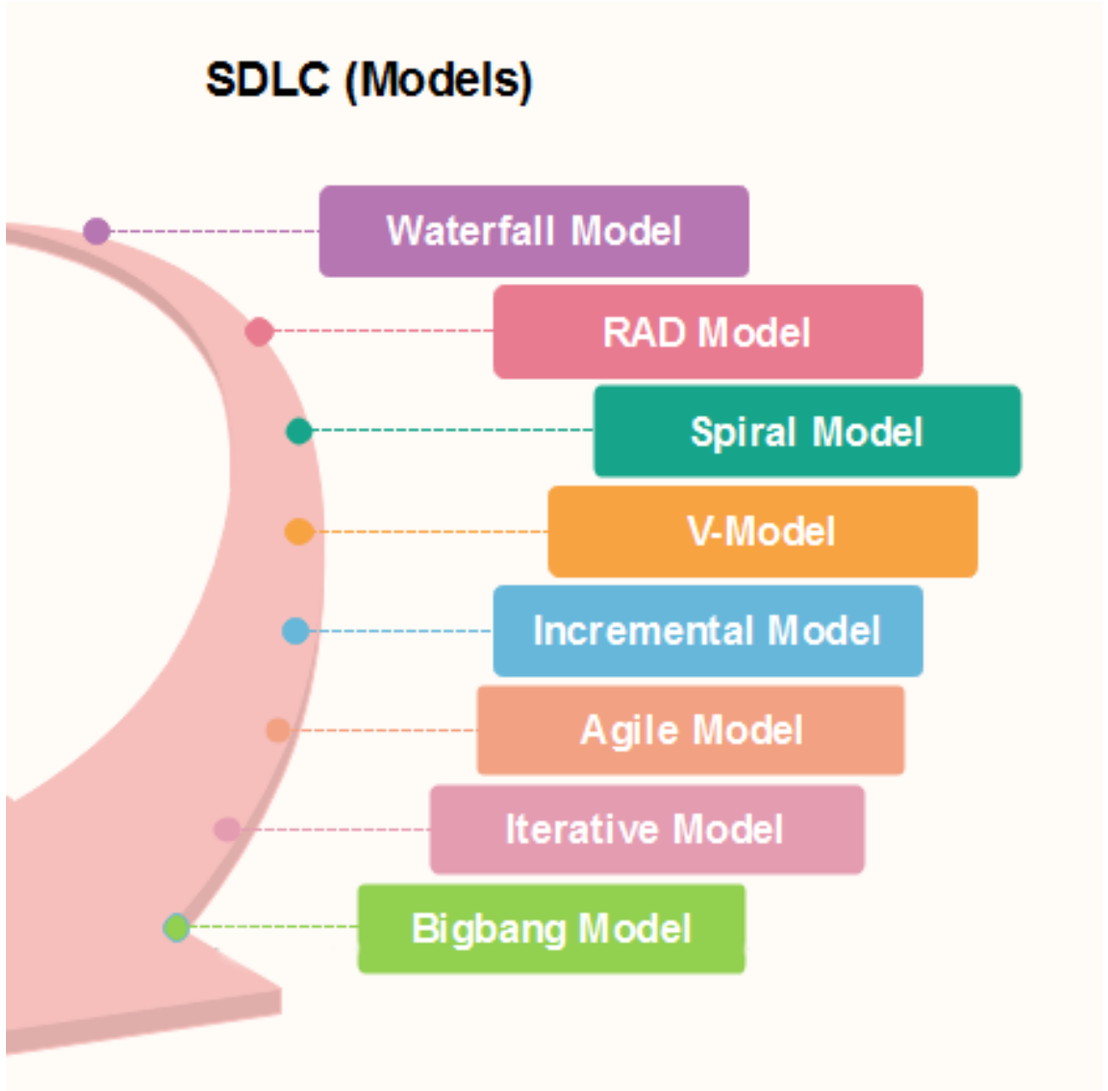
- Once **the software is certified, and no bugs or errors are stated, then it is deployed**.
- Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.
- After the software is deployed, then its maintenance begins.

Stage7: Maintenance

- Once when **the client starts using the developed systems, then the real issues come up** and requirements to be solved from time to time.
- This procedure where the care is taken for the developed product is known as maintenance.

SDLC Models

- Software Development life cycle (SDLC) is a spiritual model used in project management that defines the stages include in an information system development project, from an initial feasibility study to the maintenance of the completed application.
- There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models**."



1) Waterfall Model

- Waterfall model is the very first model that is used in SDLC. It is also known as the linear sequential model.
- In this model, the outcome of one phase is the input for the next phase. Development of the next phase starts only when the previous phase is complete.
- First, Requirement gathering, and analysis is done. Once the requirement is freezing then only the System Design can start. Herein, the SRS document created is the output for the Requirement phase and it acts as an input for the System Design.
- In System Design Software architecture and Design, documents which act as an input for the next phase are created i.e., Implementation and coding.

1) Waterfall Model (Cont..)

- In the Implementation phase, coding is done, and the software developed is the input for the next phase i.e., testing.
- In the testing phase, the developed code is tested thoroughly to detect the defects in the software. Defects are logged into the defect tracking tool and are retested once fixed. Bug logging, Retest, Regression testing goes on until the time the software is in go-live state.
- In the Deployment phase, the developed code is moved into production after the sign off is given by the customer.
- Any issues in the production environment are resolved by the developers which come under maintenance.

1) Waterfall Model (Cont..)

Advantages: Waterfall model is the simple model which can be easily understood and is the one in which all the phases are done step by step.

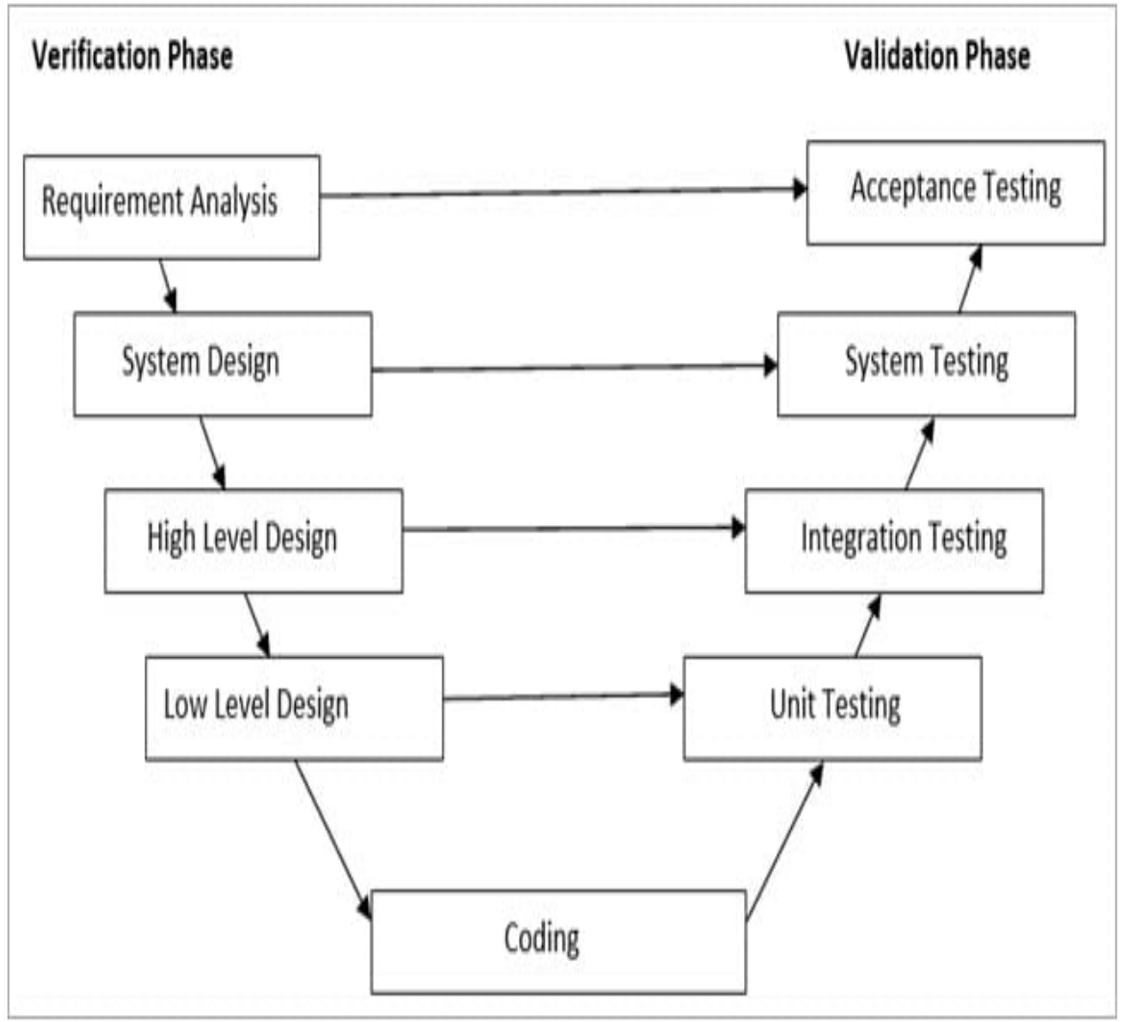
- Deliverables of each phase are well defined, and this leads to no complexity and makes the project easily manageable.

Disadvantages : Waterfall model is time-consuming & cannot be used in the short duration projects as in this model a new phase cannot be started until the ongoing phase is completed.

- Waterfall model cannot be used for the projects which have uncertain requirement or wherein the requirement keeps on changing as this model expects the requirement to be clear in the requirement gathering and analysis phase itself and any change in the later stages would lead to cost higher as the changes would be required in all the phases.

2) V-Shaped Model

- V- Model is also known as Verification and Validation Model. In this model Verification & Validation goes hand in hand i.e., development and testing goes parallel. V model and waterfall model are the same except that the test planning and testing start at an early stage in V-Model.



3) Prototype Model

- The prototype model is a model in which the prototype is developed prior to the actual software.
- Prototype models have limited functional capabilities and inefficient performance when compared to the actual software.
- Once the requirement gathering is done, the quick design is created and the prototype which is presented to the customer for evaluation is built. Customer feedback and the refined requirement is used to modify the prototype and is again presented to the customer for evaluation. Once the customer approves the prototype, it is used as a requirement for building the actual software. The actual software is build using the Waterfall model approach.

3) Prototype Model (Cont..)

Advantages of Prototype Model:

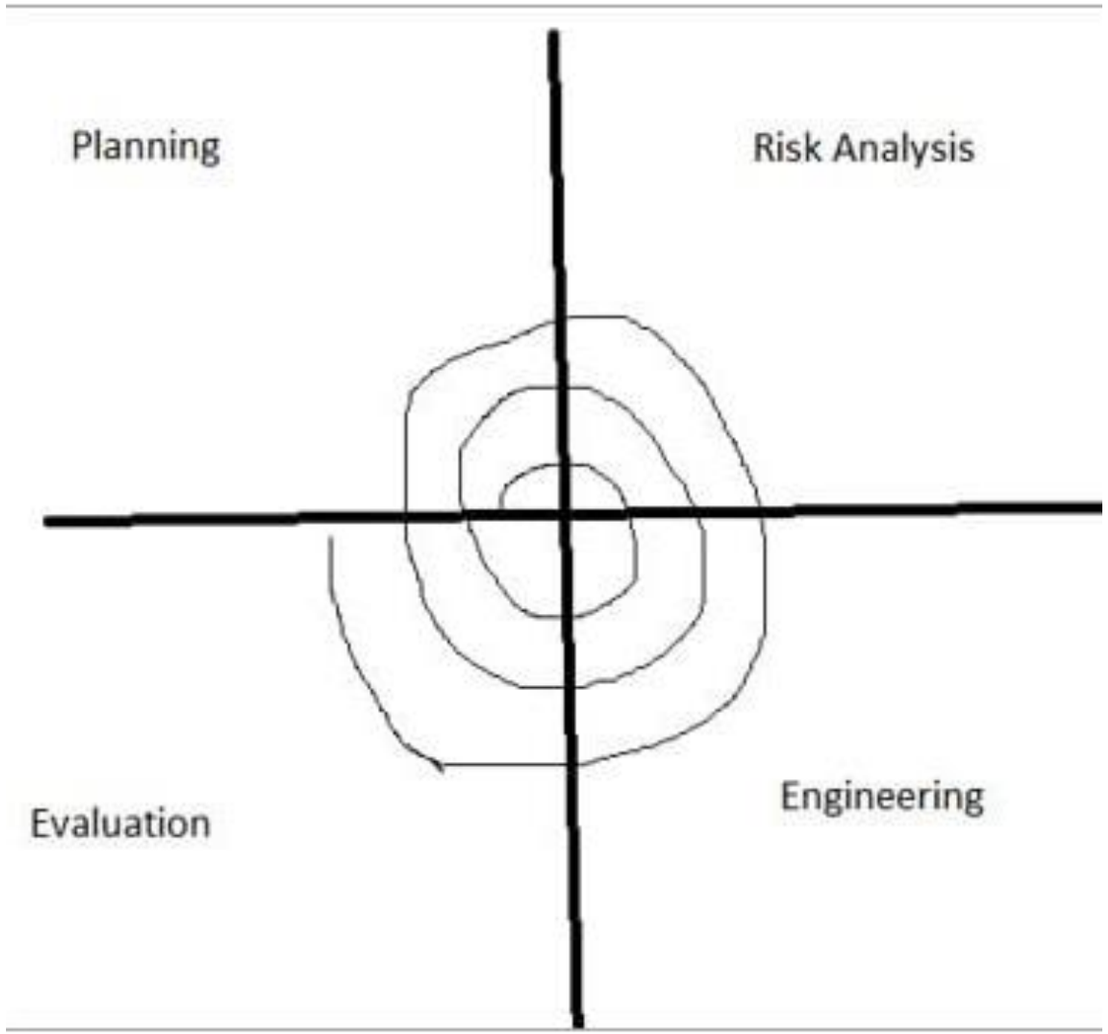
- Prototype model reduces the cost and time of development as the defects are found much earlier.
- Missing feature or functionality or a change in requirement can be identified in the evaluation phase and can be implemented in the refined prototype.
- Involvement of a customer from the initial stage reduces any confusion in the requirement or understanding of any functionality.

Disadvantages of Prototype Model:

- Since the customer is involved in every phase, the customer can change the requirement of the product which increases the complexity of the scope and may increase the delivery time of the product.

4) Spiral Model

- The Spiral Model includes iterative and prototype approach.
- Spiral model phases are followed in the iterations. The loops in the model represent the phase of the SDLC process i.e., the innermost loop is of requirement gathering & analysis which follows the Planning, Risk analysis, development, and evaluation. Next loop is Designing followed by Implementation & then testing.
- Spiral Model has four phases: Planning , Risk Analysis, Engineering, and Evaluation



5) Iterative Incremental Model

- The iterative incremental model divides the product into small chunks.
- **For Example**, Feature to be developed in the iteration is decided and implemented. Each iteration goes through the phases namely Requirement Analysis, Designing, Coding, and Testing. Detailed planning is not required in iterations.
- Once the iteration is completed, a product is verified and is delivered to the customer for their evaluation and feedback. Customer's feedback is implemented in the next iteration along with the newly added feature.
- Hence, the product increments in terms of features and once the iterations are completed the final build holds all the features of the product.

6) Big Bang Model

- Big Bang Model does not have any defined process. Money and efforts are put together as the input and output come as a developed product which might be or might not be the same as what the customer needs.

- Big Bang Model does not require much planning and scheduling. The developer does the requirement analysis & coding and develops the product as per his understanding. This model is used for small projects only. There is no testing team, and no formal testing is done, and this could be a cause for the failure of the project.

6) Big Bang Model (Cont..)

Advantages of Big Bang Model:

- It's a very simple Model.
- Less Planning and scheduling is required.
- The developer has the flexibility to build the software of their own.

Disadvantages of the Big Bang Model:

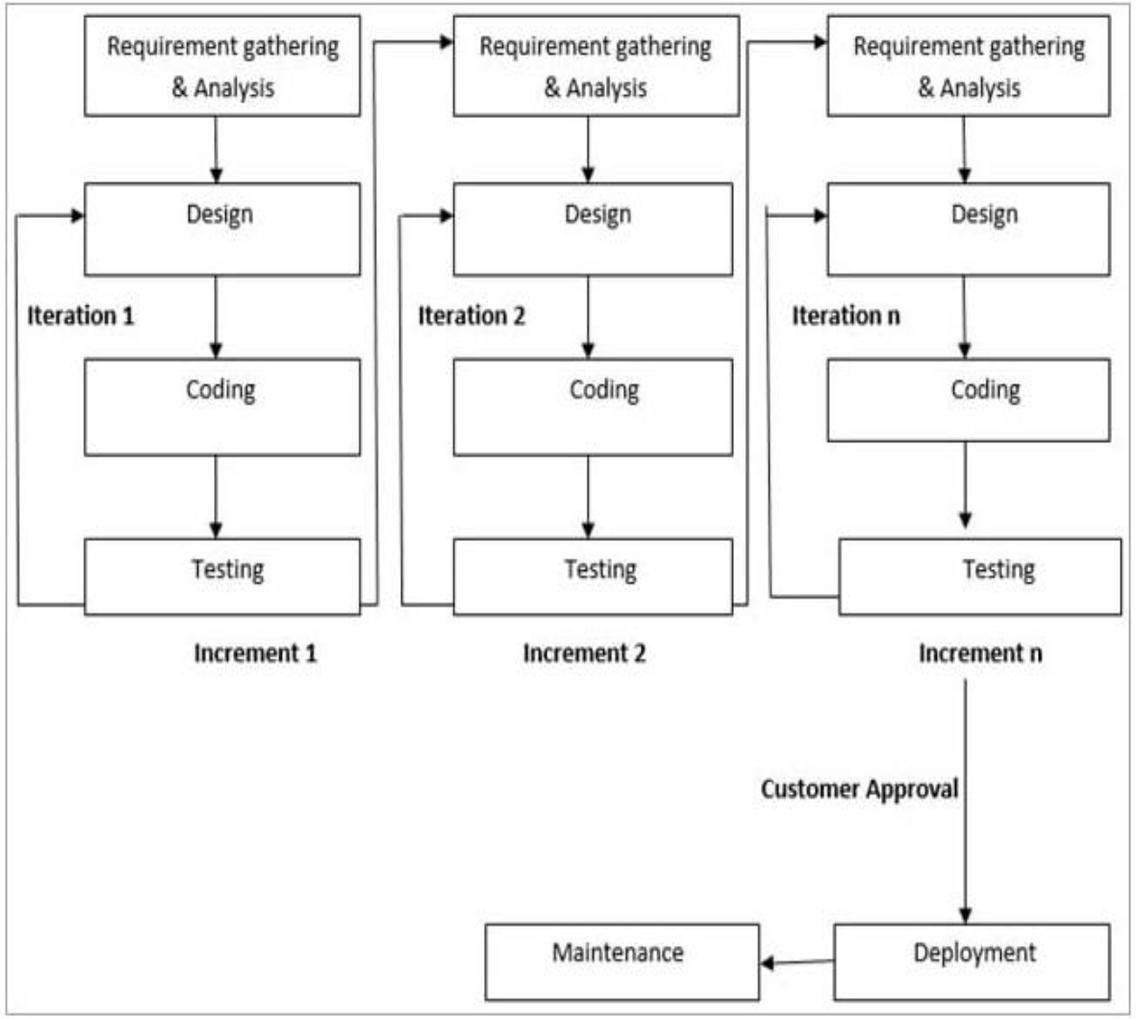
- Big Bang models cannot be used for large, ongoing & complex projects.
- High risk and uncertainty.

7) Agile Model - Values in Principles of Agile Software Development

- Agile Model is a **combination of the Iterative and incremental model**. This model focuses more on flexibility while developing a product rather than on the requirement.
- In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. Each build increments in terms of features.
- The next build is built on previous functionality. In agile iterations are termed as sprints. Each sprint lasts for2-4 weeks.

7) Agile Model - Values in Principles of Agile Software Development (Cont..)

- At the end of each sprint, the product owner verifies the product and after his approval, it is delivered to the customer.
- **Customer feedback is taken for improvement and his suggestions** and enhancement are worked on in the next sprint.
- Testing is done in each sprint to minimize the risk of any failures.



Agile Software Development

- In earlier days Iterative **Waterfall model was very popular to complete a project**. But nowadays developers face various problems while using it to develop software. **The main difficulties included handling change requests from customers during project development and the high cost and time required to incorporate these changes. To overcome these drawbacks of Waterfall model**, in the mid-1990s the **Agile Software Development model was proposed**.
- The Agile model was primarily designed to help a project to adapt to change requests quickly. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task agility is required. Agility is achieved by fitting the process to the project, removing activities that may not be essential for a specific project. Also, anything that is waste of time and effort is avoided.

Agile Software Development (Cont..)

- Agile model refers to a group of development processes. These processes share some basic characteristics but do have certain subtle differences among themselves. A few Agile SDLC models are given below:

1. Crystal
2. Atern
3. Feature-driven development
4. Scrum
5. Extreme programming (XP)
6. Lean development
7. Unified process

Agile Software Development (Cont..)

- In the Agile model, **the requirements are decomposed into many small parts that can be incrementally developed**. **The Agile model adopts Iterative development**.
- Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only.
- At a time one iteration is planned, developed and deployed to the customers. Long-term plans are not made.

Agile Software Development (Cont..)

- **Agile model is the combination of iterative and incremental process models. The steps involve in agile SDLC models are:**

1. Requirement gathering
2. Requirement Analysis
3. Design
4. Coding
5. Unit testing
6. Acceptance testing

- **The time to complete an iteration is known as a Time Box**. Time-box refers to the maximum amount of time needed to deliver an iteration to customers. So, the end date for an iteration does not change.
- Though the development team can decide to reduce the delivered functionality during a Time-box if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each Time-box.

Principles of Agile model:

- To establish **close contact with the customer** during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.
- Agile model relies on working software deployment rather than comprehensive documentation.
- **Frequent delivery of incremental versions of the software** to the customer representative in intervals of few weeks.
- **Requirement change requests from the customer are encouraged** and efficiently incorporated.

Principles of Agile model: (Cont..)

- It **emphasizes on having efficient team members and enhancing communications among them is given more importance**. It is realized that enhanced communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.
- It **is recommended that the development team size should be kept small (5 to 9 people)** to help the team members meaningfully engage in face-to-face communication and have collaborative work environment.
- **Agile development process usually deploys Pair Programming**. In Pair programming, two programmers work together at one work-station. One does code while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.

Principles of Agile model: (Cont..)

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.
- Working through Pair programming produce well written compact programs which have fewer errors as compared to programmers working alone.
- It reduces total development time of the whole project.
- Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.

Principles of Agile model: (Cont..)

Disadvantages:

- Lack of documentation.
- Agile needs experienced and highly skilled resources.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.

Thank you