Incremental Model

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.

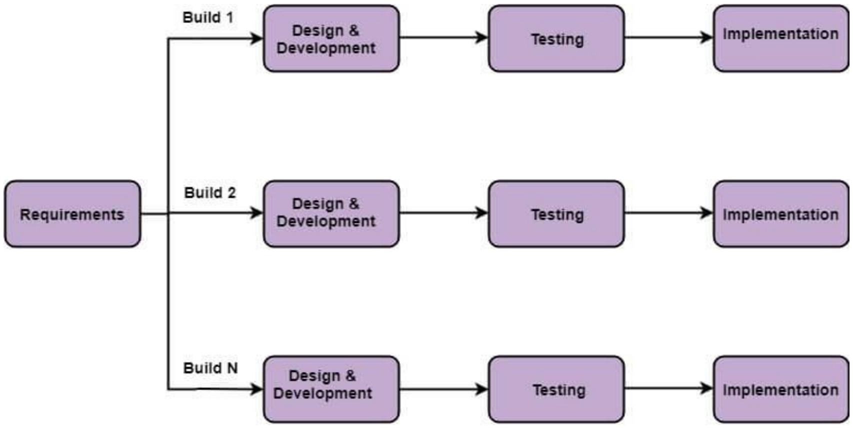

Fig: Incremental Model

The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

When we use the Incremental Model?

- When the requirements are superior.

- A project has a lengthy development schedule.
- When Software team are not very well skilled or trained.
- When the customer demands a quick release of the product.
- You can develop prioritized requirements first.

Advantage of Incremental Model

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.

Disadvantage of Incremental Model

- Need for good planning
- Total Cost is high.
- Well defined module interfaces are needed.

Iterative Model

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

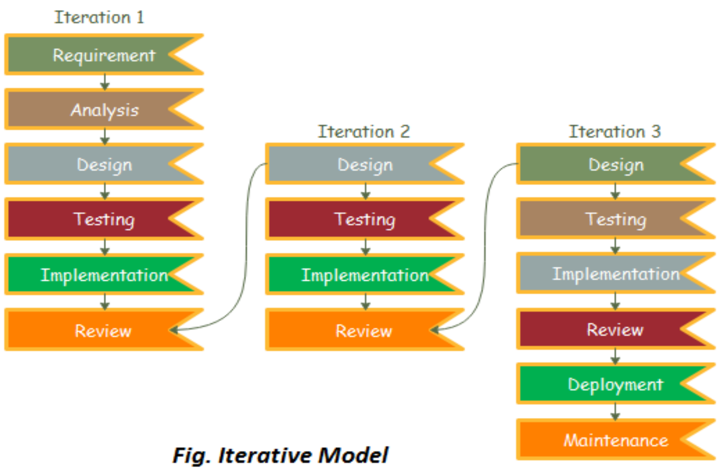

Fig. Iterative Model

The various phases of Iterative model are as follows:

1. Requirement gathering & analysis: In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.

2. Design: In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

3. Implementation: In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

4. Testing: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

5. Deployment: After completing all the phases, software is deployed to its work environment.

6. Review: In this phase, after the product deployment, review phase is performed to check the behaviour and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.

7. Maintenance: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

When to use the Iterative Model?

1. When requirements are defined clearly and easy to understand.
2. When the software application is large.
3. When there is a requirement of changes in future.

Advantage(Pros) of Iterative Model:

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

Disadvantage(Cons) of Iterative Model:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

OR

What is the iterative process?

The iterative process is the practice of building, refining, and improving a project, product, or initiative. Teams that use the iterative development process create, test, and revise until they’re satisfied with the end result. You can think of an iterative process as a trial-and-error methodology that brings your project closer to its end goal.

Iterative processes are a fundamental part of lean methodologies and Agile project management—but these processes can be implemented by any team, not just Agile ones. During the iterative process, you will continually improve your design, product, or project until you and your team are satisfied with the final project deliverable.

Example iterative processes

Engineering

Many engineering teams use the iterative process to develop new features, implement bug fixes, or A/B test new strategies. Often, an engineering team will create a few iterations that they think are equally promising, then test them with users. They'll note pain points and successes, and then continue building out the one that tested the best.

Product development

You might be surprised to realize that most product development is very iterative. Think of any personal technology you've ever purchased for yourself—there was likely a previous version before the one you bought, and maybe a version afterwards, as well. Think of the development of mobile phones throughout the years, how speakers have gotten smaller and more portable over time, or even the way refrigerators from the same brands have changed to adapt to new family needs. All of these are iterative processes.

Marketing

Some marketing teams embrace iterative processes, others not so much. But to a certain extent, a lot of marketing is iterative. For example, some marketing teams might test different advertising copy to see which one gets better engagement, or send out two versions of an email newsletter to compare click-through rates. Alternatively, a brand marketing team could use iterative design processes to identify the imagery that works best for their target audience.

Sales

Though most of a sales team's customer-facing work isn't iterative, some of their tasks can benefit from iterative processes. For example, a sales team might take an iterative approach to sending cold emails. They might have their reps send a few different email subject lines and analyze the results. Then, the team can implement the most successful subject lines moving forward.

The 5 steps of the iterative process

The iterative process can help you during the lifecycle of a project. During the steps of the iterative process, your goals and requirements will serve as the project's starting point. Then, your team will use testing, prototyping, and iteration to achieve the best possible result. Here's how:

1. Planning and requirements

During this step in the iterative process, you will define your project plan and align on your overall project objectives. This is the stage where you will outline any hard requirements—things that must happen in order for your project to succeed. Without this step, you run the risk of iterating but not hitting your goals.

2. Analysis and design

During this step, you and your team will focus on the business needs and technical requirements of your project. If step one was the process of outlining your goals, step two is when you brainstorm a design that will help you ultimately hit those goals.

3. Implementation

During the third step, your team will create the first iteration of your project deliverable. This iteration will be informed by your analysis and design, and should work to hit your ultimate project objective. The level of detail and time you spend on this iteration will depend on the project.

4. Testing

Now that you have an iteration, you will test it in whatever way makes the most sense. If you're working on an improvement to a web page, for example, you might want to A/B test it against your current web page. If you're creating a new product or feature, consider doing usability testing with a set of potential customers.

In addition to testing, you should also check in with your project stakeholders. Ask them to weigh in on the iteration, and provide any feedback.
Read: What is the Plan-Do-Check-Act (PDCA) cycle?

5. Evaluation and review

After testing, your team will evaluate the success of the iteration and align on anything that needs to change. Does this iteration achieve your project objectives? Why, or why not? If something needs to change, you can restart the iterative process by going back to step two to create the next iteration. Keep in mind that your initial planning and goals should remain the same for all iterations. Continue building upon the previous iteration until you get to a deliverable you're happy with.

If you restart the iterative process, make sure everyone is still aligned on your project goals. The iterative process can take weeks or months, depending on how many iterations you run through. Centering your iteration on your project objectives every time you restart the iterative process can help you ensure you don't lose track of your north star.

Iterative Model

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

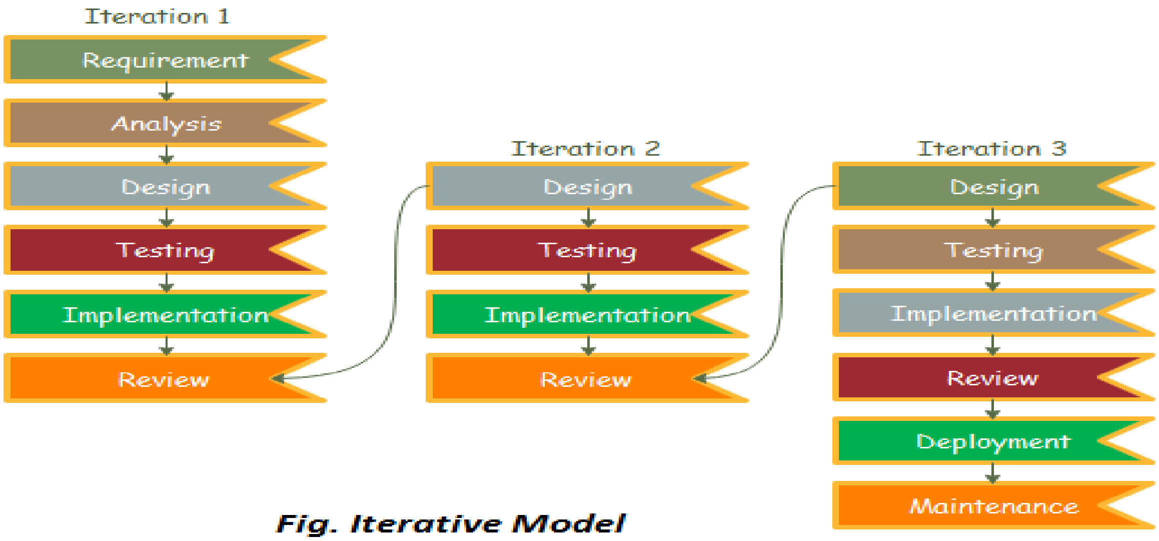

Fig. Iterative Model

The various phases of Iterative model are as follows:

1. Requirement gathering & analysis: In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.

2. Design: In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

3. Implementation: In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

4. Testing: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

5. Deployment: After completing all the phases, software is deployed to its work environment.

6. Review: In this phase, after the product deployment, review phase is performed to check the behaviour and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.

7. Maintenance: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

When to use the Iterative Model?

1. When requirements are defined clearly and easy to understand.
2. When the software application is large.
3. When there is a requirement of changes in future.

Advantage(Pros) of Iterative Model:

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

Disadvantage(Cons) of Iterative Model:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.