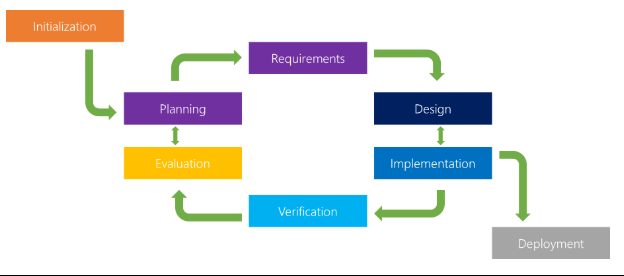
**2.0 PROJECT MANAGEMENT**

* 1. **PROJECT PLANNING**
     1. **Project Development Approach and Justification**

We have used Iterative Model for development of our project.



The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. When discussing the iterative method, the concept of incremental development will also often be used liberally and interchangeably, which describes the incremental alterations made during the design and implementation of each new iteration.

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

Phases of software development life cycle.

**Planning & Requirements**: As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

**Analysis & Design**: Once planning is complete, an analysis is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The design stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the analysis stage.

**Implementation**: With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

**Testing**: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have have cropped up.

**Evaluation**: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

# Advantages of the Iterative Model

**Inherent Versioning**: It is rather obvious that most software development life cycles will include some form of versioning, indicating the release stage of the software at any particular stage. However, the iterative model makes this even easier by ensuring that newer iterations are incrementally improved versions of previous iterations. Moreover, in the event that a new iteration fundamentally breaks a system in a catastrophic manner, a previous iteration can quickly and easily be implemented or “rolled back,” with minimal losses; a particular boon for post-release maintenance or web applications.

**Rapid Turnaround**: While it may seem like each stage of the iterative process isn’t all that different from the stages of a more traditional model like the waterfall method — and thus the process will take a great deal of time — the beauty of the iterative process is that each stage can effectively be slimmed down into smaller and smaller time frames; whatever is necessary to suit the needs of the project or organization. While the initial run through of all stages may take some time, each subsequent iteration will be faster and faster, lending itself to that agile moniker so very well, and allowing the life cycle of each new iteration to be trimmed down to a matter of days or even hours in some cases.

**Suited for Agile Organizations**: While a step-by-step process like the waterfall model may work well for large organizations with hundreds of team members, the iterative model really starts to shine when its in the hands of a smaller, more agile team. Particularly when combined with the power of modern version control systems, a full “iteration process” can effectively be performed by a number of individual team members, from planning and design through to implementation and testing, with little to no need for outside feedback or assistance.

**Easy Adaptability**: Hinging on the core strength of constant, frequent iterations coming out on a regular basis, another primary advantage of the iterative model is the ability to rapidly adapt to the ever-changing needs of both the project or the whims of the client. Even fundamental changes to the underlying code structure or implementations (such as a new database system or service implementation) can typically be made within a minimal time frame and at a reasonable cost, because any detrimental changes can be recognized and reverted within a short time frame back to a previous iteration.

We have chosen this approach as customer was not sure about the requirements of project.

* + 1. **Project Effort and Time, Cost Estimation**

Line of code=4367lines.

Effort for semidetached=3.0\*(kloc)^1.12

=3.0\*(4.3)^1.12

=15.367

Time for semidetached=2.5\*(effort)^0.35

=2.5\*(15.367)^0.35

=6.504 PM

Cost for project =7800Rs

* + 1. **Roles and Responsibilities**

Analyst: Jainil Patel

Designer: Jainil Patel

Project manager: Jainil Patel

Coder:Jainil Patel

Database Manager: Jainil Patel

Tester:Jay Patel

* + 1. **Group Dependencies**

Working in teams increases collaboration and allows brainstorming. As a result, more ideas are developed and productivity improves.

Two or more people are always better than one for solving problems, finishing off difficult tasks and increasing creativity.

Everyone is unique and has different skills, backgrounds and experiences. Therefore, others in a team can help you see things from a different angle.

Teamwork encourages communication between team members. For this reason, relations between employees tend to be better and over time employees learn to communicate better.

In some teams, there may be members who sit back and let others do all the work. In these types of teams conflicts may occur and this can affect the mood of others in the team.

Working in a team requires many meetings and these meetings, if not managed well, can go off topic and decrease the efficiency of the team.

Making decisions can take longer for the sake of finding a consensus.

* 1. **PROJECT SHEDULING**



