**SOFTWARE DEVELOPMENT MODEL**

We have chosen ‘Big Bang’ model as our software development model. This model aims at delivering the project at one shot without any incremental development. The Big Bang model is SDLC model where we do not follow any specific process. The development just starts with the required money and efforts as the input, and the output is the software developed which may or may not be as per customer requirement.

Big Bang Model is SDLC model where there is no formal development followed and very little planning is required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis.

Usually this model is followed for small projects where the development teams are very small.

Big Bang Model design and Application:

Big bang model comprises of focusing all the possible resources in software development and coding, with very little or no planning. The requirements are understood and implemented as they come. Any changes required may or may not need to revamp the complete software.

This model is ideal for small projects with one or two developers working together and is also useful for academic or practice projects. It’s an ideal model for the product where requirements are not well understood and the final release date is not given.

Aptness of the model with our project:

The main of our project is to simulate a wireless sensor network with sufficient number of nodes and implement the fuzzy detection algorithm to detect the infected nodes. Thereafter, twin ball rolling technique is made use of in order to retrieve the packets that are trapped in the infected area and re-route them to the destination. Since there is no much scope for incremental development of the project and since this is an academic project, we strongly feel that this approach is the apt one to be followed.

Pros:

* This is a very simple model
* Little or no planning required
* Easy to manage
* Very few resources required
* Gives flexibility to developers
* Is a good learning aid for new comers or students

Cons:

* Very High risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Can turn out to be very expensive if requirements are misunderstood