3. Research Methodology

3.1 Introduction to software development methodology

As per the definition of Madsen [2] software development process had changed a lot and the growth across the recent times is rapid and also includes various techniques and methods and they are implemented across the overall software development process. Software development process can be considered as a complex process as it involves lots of steps towards implementation and a standard life cycle steps are followed across this process. There are different types of software development models and most of them are proved to be successful and there are some failures even. Across the overall software development life cycle requirements gathering can be considered as the vital step and this is the phase where most of the software development models fail.

As per the opinions of Sanjay [25] in the software development process the requirements gathering phase is happened to be typical task for the companies which are developing the software methods. The requirements gathering phase involves the different steps in which gathering the client requirements is the first step, meeting with the client requirements is the second phase and satisfying the clients requirements is the final step in this process. Most of the software research
methods will face failure because miscommunication process will take place among client and developer.

As per the opinion of Korson [26], most of software development models fail at the requirement gathering phase due to miscommunication and lower interaction levels and even there are lots of cases where the projects fail in middle and finally the clients and vendors need to face issues. Requirements gathering phase has the key role to play across the overall software development process and there would be many hidden issues and it is always required to make them clear before freezing the requirements. It is the responsibility of both the vendors and clients to make sure all the requirements are discussed at this stage and if anything is missed, the overall design of the project should be changed [27]. Apart from lots of efforts consumed across requirements gathering process, there are many chances for issues and thus a perfect implementation tools are required in this context to avoid these mistakes.

As per the definition of Vliet(2008), requirements gathering phase has the key role to play across overall software development and it is always required for a separate area against the requirements engineering and it includes various methodologies and frameworks[28,29]. All these techniques should be perfectly implemented across the requirements gathering phase and thus to avoid all the mistakes possible at this phase and thus implemented the desired software development model.
3.2 Software Development Lifecycle Models

As per the opinion of Cloete (2011), all the phases for software development cycle are required against developing perfect software and all these phases are executed in a sequential manner. In general there are different types of software development model and companies follow these models as per their own requirements[24,30]. In this section the most common phases of software development are discussed. There is always a connection from phase to another phase across the software development process and the output of one phase will be input to the next phase. The very first step across software development process is requirement gathering and it plays the vital role. Once all the requirements are gathered and freeze, they are translated to a raw design phase. Ample design is generated across this phase and once the design is ready it is taken towards implementation phase. The required business logic is developed using the code and once the coding is done, it is sent for testing phase. Once the testing is done the corresponding product is delivered to the user. There are many software development models and few of them are discussed as below.

3.2.1 Waterfall Model

As per the definition of Korkmaz(2007), waterfall model is the typical and traditional models among all the software development models and it follows the
liner sequential lifecycle process and it can be considered as the easy process to implement[31]. Every phase is started once the previous phase is completed and thus a sequential method is followed. The most important advantage with this model is that, at end of each and every phase a detailed evaluation of the project is provided and a detailed analysis can be done against the overall success of the phase. It can be observed that the phases across water fall model never overlap.

Figure - 1: Waterfall Model
3.3 V-Shaped Model

As per the opinion of Kothari (2010), V-shaped model can be considered as another lifecycle model which is almost similar to waterfall model. Different sequential paths are executed across this process similar to waterfall model and even in this model each phase is started once the previous phase is completed[32]. The main step across this model is the testing phase, where the main concentration is given to requirements gathering phase across waterfall model.

Requirements gathering are the very first phase even in this model and when the requirements are done, a clear test plan is provided against the requirements gathered. Once the core functionalities are defined across the requirements gathering phase, a detailed test plan for these functionalities is developed and thus the level of development requirements are understood. Design phase across this model is divided into two levels like high level design phase and low level design phase [33].
As per the opinion of Ghezzi (2004) it is clear that the architecture and design of the software system mainly concentrates across the high level design. All the possible fragments of the software system being developed are considered and the level of integration possible is tested across integration testing and an integration test plan is created after the design is done[34]. The actual design is done at the low level design phase and once it is done unit test cases are done and they are used across the testing phase. Coding and implementation is done once the design and unit test cases are done, an execution path is developed and thus across the shape-V the required test plans are created and implemented once the coding is done.
3.4 Structured evolutionary prototyping model

A basic prototype is created across this model once all the requirements are gathered and the created prototype is evaluated by the clients and based on the feedback from the client the prototype is changed. Developers and clients need to maintain a perfect communication in this model, as the prototype need to be critically evaluated. This process is repeated till the client or end user is satisfied and once it is done the prototype is finalized and the product is developed [35]. There are various evolutionary steps in this context and they are as given below

- A basic project plan is developed initially
- High level paper prototype is developed
- All the raw requirements are analyzed
- Prototype is created based on the basic attributes
- Required database, algorithms and user interface screens are designed.
- Developed prototype is demonstrated and send to user for evaluation
- This process is repeated till the end user is satisfied (Clayton, 1995).

3.4.1 Structured Evolutionary Prototyping Strengths

The main advantage with prototype model is that the customer can understand all the requirements and even the developers can gain the required knowledge based on the customer requirements. This model is more flexible for both the design and
development and the requirements gathered can be easily analyzed based on the customer feedback. All the additional requirements based on the customer feedback and they can be added across the design phase.

3.4.2 Structured Evolutionary Prototyping Weaknesses

As per the opinion of Griffith (2008), the main disadvantage with prototype model is that code-and-fix is not all accurate and there are chances for the reputed quick-and-dirty models. When it comes to maintenance phase few important issues may be bypassed across this model[36].

3.4.3 When to use Structured Evolutionary Prototyping

The main usage of prototyping model can be analyzed across the requirements clarification and thus when it is required to improve the user interface. Prototype model provided the real demonstration of the products in prior to final release and thus the developers and customers can have ample idea in this context and always an object oriented development is done at this model. All the important phases where prototype model is applied are discussed as below

- When the requirements are unclear and unstable
- At the requirements clarification phase
- While developing the user interface
• When a short lived demonstration is required to the customer
• While developing a new and original development plan
• When the system requires object oriented design (Jhonson, 2009).

3.5 Rapid Application Model (RAD)

As per the opinion of Langer (2008), the key process included across Rapid Application Model (RAD) is as discussed as below

• A detailed workshop is conducted and the overall business process is discussed across the requirements planning phase
• All the required automated tools are reviewed and the user requirements are analyzed across the user description phase.
• Construction phase is related to production purely and all the production tools are reviewed to generate the code and screens etc.
• User acceptance testing is done at cutover phase, where the user acceptance is gathered while delivering the actual product[37].

3.5.1 RAD Strengths

There are many advantages with RAD software development model and few of them are discussed in this section. The main advantage with model is that it requires lower human resources and thus the overall costs are reduced. Time box
approach leads to reduce the overall costs and also the risk involved while scheduling the project. The other advantage with this model is that there is a continuous interaction with customers throughout the software development. Documentation is reduced a lot and coding is given more priority in this mode and all the requirements are gathered using modeling (Kernzer, 2009)[38].

![Figure 3: RAD model](image)

**Figure - 3: RAD model**

### 3.5.2 RAD Weaknesses

The key advantage with RAD is discussed in the previous section and apart from these advantages there are few limitations and they are discussed in this section. As per the opinion of Goertzel (2007), the main limitation with this model is that it is always hard to apply for legacy systems. When the level of risk is analyzed the modularized systems are more prone for skipping the risk analysis and thus the customers need to face the risk finally.
3.5.3 When to use RAD

The main usage of RAD can be analyzed when all the requirements are gathered and developer need to involved across all the lifecycle stage of software development. Time boxed requirements can be analyzed across the implementing RAD. This model is used when the performance of the system is given lower priority and also when the technical risks are ignored across the process of modularized.

3.6 Introduction to Agile

As per the opinion of Attarzadeh(2008), agile can be defined as the light weight process and it can be used widely across the project management. It has lots of differences when compared to the other project management techniques and it involves the role of project manager to define the tasks to all the team members and thus take the required control over the project. Team involvement is always required in this model and the deadlines are decided over the team meetings that were conducted across this model[39].

Project manager is involved across developing the high level project plan and this can be considered as the main difference when compared to other models. The actual start of project and work is done based on the high level plan developed and thus all the long term requirements are analyzed and they are broken down in this
model. Solution to the project is created based on high level vision where iteration based approach is followed by the project managers across the software development process. A new model is developed based on the iteration results achieved from the previous model across Agile and the required process description is given as below (Wysochi, 2003).

**Figure - 4: Agile Methodology**

As per the opinion of Cohn(2004), it is clear that the main advantage with agile project management is that, there is always required feasibility for the team
members to develop their own work plan irrespective of the plan followed by the project managers. The level of collaboration among the stakeholders, customers, team members and top level project managers is streamlined with the implementation of agile project management techniques[40]. A proper communication channel can be maintained when there are some changes across the requirements and the roles of the team members can be clearly defined by the project managers using this mode.

3.7 Methodology Used

A clear discussion is done against different software development models are done in the previous section and the actual methodology followed in this project is given in this section. Waterfall model is followed in the simulation process and the main aim of this project is to evaluate the performance of AODV under black hole effect.

3.8 Defining methodology

In general methodology can be defined as the process of gathering the available resources to develop a research and following the require procedures while solving the research problem. There are different types of research methodologies and few of them are as discussed below
3.9 Types of Research Methodologies

There are different types of research methodologies and among them qualitative and quantitative research methodologies are discussed as below[3,41].

3.9.1 Quantitative research method

Quantitative research methodology is used in this research and it includes all the required articles and also a case study as well. The main topic considered is intruder detection using the data mining systems and thus internet resources are used to gather the required information. A typical theoretical based approach is used at this level and the required empirical analysis is done. The required primary data is gathered from different sources and they are given as below

3.9.2 Primary Data

- Journals
- Articles
- Books
- Web materials
- Case studies
3.9.3 Source of Primary Data

The required primary source of data is gathered using the case studies, articles, journals and websites and this information is analyzed for the research. All the required articles, books and case studies are analyzed and the required information is gathered as per the requirements of the research problem considered [42].

3.10 Ethical Issues

There are few ethical issues involved across this research and they are discussed in this section. Authentication may be required to gather the information from different articles and journals and this can be considered as the primary ethical problem. As the main aim of this project is to develop intruder detection system based on data mining concepts, it may be required to interact with different network administrators and thus while gathering information from them the ethics need to be followed.