CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION

Requirements engineering (RE) refers to the process of formulating, documenting and maintaining software requirements and to the subfield of Software Engineering concerned with this process. It is categorized into a source and contractual applicability. In the requirements engineering, a bridge is constructed according to the system requirements. Requirements engineering grants permission to a requirements engineer for examining the performance of the context of the software work. The stakeholder’s requirements are engaged in the software requirement specification (SRS). A careful evaluation and prediction of valid requirements are required for better SRS. The information, the function and the behavior of the collected requirements make a deep impact on the resultant design. The seven distinct tasks of the requirements engineering are inception, elicitation, elaboration, negotiation, specification, validation and requirements management. A compliance level of the requirement engineering is determined by using the depth of the compliance mandated for a requirement. The tasks performed in the requirement engineering are:

i. Requirement Elicitation: It is also called Requirement gathering. The requirements of the customer and the developer are identified and articulated during this process. The needs of the user are used to determine the stakeholder’s requirements. This elicitation is accomplished through two activities such as collaborative requirements gathering and quality function deployment.

ii. Requirement Analysis: It is also called Requirements Engineering. It is the process of identifying user expectations for a new or modified product. A set of agreed requirements is specified during the requirement analysis to achieve the requirement contract. It primarily focuses on the development of a refined technical model of software function, features and constraints. These requirements are called functional specifications.
iii. **Requirement Specification**: It is an inclusive document that details the intended purpose and milieu for software under development. The developed document distinctly and accurately records each of the software requirements. The requirement specification describes the function and performance of the computer-based system. The requirement specification ensures the user requirements to provide conformation to the document standard. A requirement engineer is responsible for this activity.

iv. **Requirement Validation**: The requirement validation is a required basis for the architectural design. Requirement Validation addresses key issues namely, completeness and cohesiveness of functional coverage, Consistency of user requirements, Traceability i.e., each requirement is traced back to its stakeholder, coverage of non-functional parameters and testability of functional and non-functional requirements. The members in this task include software engineers, customers, users and other stakeholders.

v. **Requirement Management**: Requirement Management is the process of planning and controlling the requirements elicitation, specification, analysis and verification. This management task performs a group of activities to identify, control and track all the activities in Requirement engineering.

Some of these tasks occur in parallel and are adapted to the software model requirements. These elements serve to establish a solid foundation for the design and construction of the software. This task is typically performed in an iterative, incremental, concurrent and time-boxed manner. In requirements engineering, the requirements engineer analyzes the context of the software work to be performed and observe the information, function and behavior of the requirements. A requirements engineering process is developed by performing scope management, requirements verification and requirements configuration control.

### 1.2 SOFTWARE REQUIREMENTS

Software requirements defined in IEEE standard glossary of software engineering terminology, primarily deals with the establishment of the stakeholders’ needs. The software development cycle is defined in the desired system requirement document. The requirements describe the behavior and activities of the software. The requirements are
classified into several levels, such as business, user and functional requirements and various non-functional requirements.

Figure 1.1 shows the relationship of the software requirements. The requirements are gathered from the business manager to define the quality attributes. The system requirements, the functional requirements and the non-functional requirements are used to determine the software requirement specifications according to some certain constraints.

- **Business requirements**: It establishes high-level requests from the user or developer to develop a system. The business requirements for a software application are defined by the managers. It helps the company to operate or compete efficiently and successfully in the marketplace. The business environment describes the globalization, customer satisfaction, parallelization, agility, virtual enterprise and Total Quality Management (TQM).

- **User requirements**: It specifies what the user expects the software to be able to do. It is linked to the quality attributes to improve the description of the product functionality. It is improved by demonstrating the characteristics of a product in various dimensions. This is more important to users and developers.
• **Functional requirements**: It describes the interaction between the system and its environment. It also specifies the functionality and behavior of the system. Here, the input and output of the software are considered as the functional requirements. These functional requirements are supported by non-functional requirements.

• **Non-functional requirements**: It describes the limitation and restriction of the system. These requirements contain standards, regulations and contracts. While designing the software product, the developer has many constraints or restrictions to elaborate the performance characteristics of the system. The non-functional requirements are differentiated from the functional requirements.

The software requirements not only describe the information flows, but also focus on the constraints of the system functions. The requirement provides high quality to the persuading developers. Some of the features and characteristics of this requirement are described below,

• **Completeness**: All probable states, changes and constraints of the requirements are mentioned to reach a complete state. Each requirement fully describes the functionality and contains the necessary information for the developers to design and implement the functionality.

• **Strict accuracy and correctness**: The user and the developer accept the requirements to disclose the error. These characteristics of accuracy and correctness are mainly considered as the quality of the requirements. This stage focuses on the precision and inconsistency of the requirements.

• **Feasibility**: It is an analysis and evaluation of a project based on the user requirements. The feasibility factors such as legal feasibility, operational feasibility, economic feasibility, technical feasibility and schedule feasibility are considered.

• **Consistency**: The incompleteness theorem is mainly used to provide better consistency. A set theory is used to describe the relationship between the consistency and the completeness.
• **Verifiable:** It refers to a set of activities to ensure the accurate implementation of the software with some specific function. The verification approaches, i.e. inspection or demonstration plays a vital role to determine the proper implementation in the software product.

• **Traceability:** It refers to the ability to trace a design representation or an actual program component back to the requirements.

• **Necessity:** The user provides a conformation to an external system requirement or standard.

### 1.3 STAKEHOLDERS REQUIREMENTS

A stakeholder is an organization or individual who can affect or be affected by the actions of a business. The categories of the stakeholder include end users, managers and others involved in the organizational processes. The primary objective of the requirements engineering is to propose the concepts of user and system requirements and to describe the functional and non-functional requirements. The requirements are classified into different types namely:

• Architectural requirements

• Business requirements

• Stakeholder requirements

• Functional requirements

• Quality-of-Service requirements

• Implementation requirements

The requirement elicitation process is accelerated to gather the requirements of the stakeholders. The stakeholders are gathered together for an intensive meeting to collect the requirements of the stakeholders.

Figure 1.2 shows the gathering of requirements from the stakeholders by conducting the meeting. The following steps are performed during the meeting to manage stakeholders:
i. Identify the stakeholders to identify and collect the requirements.

ii. Identify the document needs of the stakeholder.

iii. The influence or interest of the stakeholder is analyzed to know the stakeholder expectations.

iv. The stakeholder expectations are managed.

v. The management takes necessary action as per the stakeholder’s expectations.

vi. Finally, the review and repeat process is performed during the stakeholder management.

Figure 1.2 - Stakeholder Management

The stakeholder management determines the nature or legitimacy of the stakeholder, stake group power and specific groups within generic groups. It creates the opportunities to build better productive working relationships with the stakeholders. The effective capabilities of the stakeholder management are at a rational level, a process level and transaction level. The principles involved in the stakeholder management are to monitor, listen, communicate, adapt, recognize, work, avoid and acknowledge the conflict.
1.3.1 STAKEHOLDERS DATA PREPROCESSING

A preprocessing is performed in the real world data because the data is either incomplete or noisy or inconsistent. In the incomplete data, the data lacks attribute values or aggregate data. In the noisy data, the data contain errors or outliers. In the inconsistent data, the data contain conflicts in codes or names. The quality decision is made depending on the quality of the data. An information preprocessing system is used to identify and discover the issues during the requirement analysis. Normally, the requirement preprocessing involves in cleaning, normalization, transformation, feature extraction and selection. The requirement preparation creates the major issue for both data warehousing and data mining.

The reported errors, the unusual values and inconsistencies in the data are recorded in the database system for transaction. The updating, deleting or transformation of the training set data is referred by using the data preprocessing techniques. A cell segmentation data are introduced during the data preprocessing. The surrogate variables are created during the feature extraction process. The data transformation and the input variables are selected for increasing the accuracy of the intelligent model to be developed. The data mining defines the basic concepts of the data warehouses and Online Analytical Processing (OLAP). The data warehouse is a relational database, which is used for designing the query and analysis instead of the transaction processing. In the data warehouse, the historical data are derived from the transaction data. A data warehouse environment consists of an extraction, transformation, and transportation, loading solution, an OLAP engine, client analysis tools and other applications.

In Figure 1.3, the following categories organize the major tasks performed in the data preprocessing: data cleaning, data integration, data transformation, data reduction and data discretization.

The extraction, integration and aggregation processes of the needed resource are performed on the data from the operational resources. A schema extraction and translation process is performed in the data extraction. A schema matching process is performed in the data integration. A schema implementation is performed in the data aggregation. These are recorded in the data warehouse. A scheduling, logging, monitoring, recovery and backup are maintained by performing the data extraction, data transformation and data loading. The preprocessing procedure includes the following activities:
**Figure 1.3 - Data Preprocessing**

- **Data cleaning**: The data cleaning process is used to remove the missing values, smooth noisy data, outliers and inconsistency in the data. The data cleaning consists two-step process namely error detection and error correction in the data set.

- **Data integration**: The combination of technical and business processes from disparate sources into meaningful and valuable information is known as the data integration. Multiple databases, data cubes or files are integrated with each other.

- **Data transformation** - Data transformation is the process of converting data or information from one format to another, i.e. the format of a source system into the format of a destination system. The two key phases of the data transformation are data mapping and code generation. In the data transformation, the normalization and aggregation are performed.
• **Data reduction**: The transformation of numerical or alphabetical digital information is empirically or experimentally derived in the data reduction. The same or similar analytical results are produced by obtaining a reduced representation of the dataset. These contain data aggregation, attribute subset selection, dimensionality reduction and numerosity reduction.

• **Data discretization**: The data discretization is a process of transforming the quantitative data and the qualitative data in data mining application. In the data discretization, the data is sorted in ascending or descending order according to the discretized variables between the data sets.

A randomized path planner uses the unclear, undecided and unidentified requirements, the incomplete requirements and ambiguous or contradictory requirements of the software. The randomized path planner primarily describes a particular plan without considering the other options to achieve the main objective. The randomized path planner is utilized for the requirement analysis and system design. A filter process is performed in the requirement engineering to avoid the unwanted requirements. The issues are declared as the unclear requirement (UR). In several cases organization and stakeholders are unable to identify the issues of the unclear requirements without knowing the development process demand. The information preprocessing system is used to gain the requirements of the development process in both UR and Incomplete Requirement (IR). In the Ambiguous or Contradictory Requirement (AR/CR), the ambiguity is avoided by the developer during the development process. The functional and non-functional requirements are analyzed by the developer to avoid ambiguity in the development process by using the following activities:

• To provide clear objectives and define the goal point,

• To set the focus point and provide clear task results

• To describe the framework of the process model
1.4 FEATURE SELECTION

Feature selection is also called variable selection or attribute selection or variable subset selection. During the feature selection process, the dimensionality reduction is performed to reduce the cost of the features. It is very safe and inexpensive to collect a reduced set of predictors. A backward or forward stepwise method is used to select the optimum number of features. The vital role of this feature selection is to obtain number of features that yield a better classification result. During the feature selection, the entire space is searched for possible feature subset. The searching strategy may be an optimum, heuristic and randomized. Once selected, any one of the feature subset is used to check whether it is optimal or near-optimal according to a certain criterion to build effective predictive models. It is mainly performed to remove the noisy features such as irrelevant and redundant features available in the requirements. This removal, improves the performance and accuracy of the feature classification. The following are the benefits of the feature selection:

- Improving the model interoperability
- Shortening the training times
- Enhancing the generalization by reducing over-fitting
- Decreasing the computational time
- Improving the performance analysis capability
- Providing the high dimensional space for the classifier
- Reducing the complexity problem.

In this feature selection, a subset of the original features is selected. An evaluation criterion is used to evaluate the optimality of a feature subset. The number of features is increased to expand the dimensionality of the domain. A typical feature selection process consists of subset generation, subset evaluation, stopping criterion and result validation. The subset generation, i.e., the search procedure produces candidate feature subset for evaluation depending on a specific search strategy. Each candidate subset is measured and compared with the existing one corresponding to a certain evaluation criterion.
Figure 1.4 shows that the key steps for feature selection process. In the feature extraction process, if the newly generated subset is better than the existing subset, then it is replaced by the existing one. The subset generation and evaluation process is repeated until the given terminating criterion is satisfied. The selected subset is validated with prior knowledge or different tests through fake and/or real world datasets. The feature selection is found in several areas of the data mining such as classification, clustering, association rules, regression.

For illustration, the feature selection is also called subset or variable selection. The feature selection algorithm mainly focuses on classification and clustering. A supervised feature selection mainly focuses on the feature selection for classification with the labeled data. In an unsupervised feature selection, it mainly focuses on the adaptation of general procedure with the feature selection for clustering with unlabeled data. According to the evaluation criteria, the three types of feature selection algorithm are: Filter, Wrapper and Hybrid algorithms. In this proposed approach, the Hybrid algorithm is used for feature selection in an efficient manner.
1.5 FEATURE CLASSIFICATION

A classification is performed according to the virtual content of the data. It mainly refers to the task of extracting the information from the database. It is used to address the growing or constantly changing collection of the unstructured data. The important steps for classification are to identify data owners, define data of interest, use meta-data to focus, accelerate and report data. The issues are automatically identified during the identification of data owners. This process is a key part of the classification process. A feature classification uses a knowledge-based approach to perform requirement analysis to detect potential problems and primarily lack of testability.

A classification scheme is included with the knowledge-based system to represent the software requirements as factual knowledge. This classification scheme is mainly used to index the requirements. The indexed requirements are listed with better software portability and operating efficiency. A knowledge-based approach is mainly used to detect the conflicts, incompleteness and ambiguity of the requirements. Different algorithms are used for feature classification such as Support Vector Machine (SVM) classifier and Artificial Neural Network (ANN) classifier.

- **Support Vector Machine classifier**: The SVM is a machine learning tool that is mainly used for classification and regression depending on the supervised learning. The primary objective of this machine is to construct the function that can accurately predict relationship of the stakeholder requirements. It is an efficient algorithm for high dimensional data and controls tradeoffs between classifier complexity and error. But using this classifier for several requirements creates high algorithmic complexity and extensive memory requirements.

- **Artificial Neural Network Classifier**: It is mainly used to classify the preprocessed requirements. In this classifier, the feature analysis is performed. For that, the preprocessed requirement is classified into a training module, a verification module and an application module to improve computational efficiency. It contains multiple training data, so that the ANN classifier cannot be easily applied because of hard and rapid rules.
In this proposed approach, the fuzzy algorithm is used for requirement classification. The fuzzy algorithm is one of the most used fuzzy partitioning schemes. The classification of the data samples is performed by an unsupervised learning algorithm without utilizing the class label information. In this algorithm, the global stochastic tool is implemented to solve different function optimization problem.

1.6 NETWORK FORMATION

A network describes a collection of nodes and links. The routing strategies of the network are fixed routing, virtual circuit and dynamic routing. The connection strategies of the network are circuit switching, message switching and packet switching. The seven layers, i.e. Open Systems Interconnection (OSI) layers such as physical layer, data-link layer, network layer, transport layer, session layer, presentation layer and application layer are located in this network structure. In the network formation, the network structure plays a significant role in determining the result of several cost-effective associations. A network formation process is analyzed in a dynamic framework. The network structure is mainly used to connect various individuals and the network formation is performed based on the decisions of several participants. The key properties of the network structure are defined as the degree distribution, the clustering and the average distance.

A linking activity is performed by the network formation. The key issues in the network formation, such as payoffs, power and information occur in the network formation modeling. The network formation process, i.e., path dependent determines the coverage to an efficient network structure depending on the information. A stochastic simulation algorithm also called Monte Carlo algorithm is a well-known Bayesian inference algorithm. The principles involved in the network formation are named as reciprocity, popularity and triadic closure. The stochastic sampling algorithm is efficiently reported as an adaptive importance sampling for Bayesian network to perform a requirement clustering. In this proposed work, a Bayesian network is used for network formation.

1.7 CLUSTERING SIMILARITY

The primary goal of the cluster is to determine the intrinsic grouping in a set of unlabeled data. The clustering is the assignment of data objects into groups, so that the data objects from the same cluster are more similar to each other than objects from
different clusters. Commonly used clustering algorithms are K-means, Fuzzy C-means, Hierarchical clustering and mixture of Gaussians. The hierarchical clustering is further classified into an agglomerative hierarchical clustering and a divisive hierarchical clustering.

- **Agglomerative hierarchical clustering:** This algorithm works by grouping the data one by one on the basis of the nearest distance measure of all the pair-wise distance between the data point. Finally the distance between the data point is recalculated.

- **Divisive hierarchical clustering:** In this clustering, the data objects are clustered in a top-down manner. At first, all the data objects are assigned to a single cluster. Next the cluster is divided into two least similar clusters. Finally, proceed recursively on each cluster until there is one cluster for each observation.

In the hierarchical clustering, the Jaccard algorithm is used for similarity clustering. The k-means algorithm is a numerical, unsupervised, non-deterministic iterative method, simple and rapid. It is used to solve the problem of the well-known cluster. The non-deterministic iterative method is effectively proved to produce better clustering results. The clustering result is mainly suitable for creating the globular clusters for improving the efficiency of the k-means algorithms by the researchers. The symbol attribute of the data is handled by a partitioning clustering algorithm to find the initial cluster centers.

The k-means based clustering algorithm achieves the pattern assignment changes to the clusters. The k-means clustering uses a heuristic to determine the changes of the protocol pattern by using a simple check. If the changes do not occur, then the distance calculations are not needed. During the process of consecutive iterations, the cluster centroids are small. In the k-means clustering, the initial centroids i.e. mean points, are randomly chosen in the cluster. The closeness of the cluster is measured by using a Euclidean distance, cosine similarity, correlation and so on.

The k-means algorithm is a popular data-clustering algorithm. It requires the number of clusters in the pre-specified data. The k-means algorithm is implemented in the data mining or data analysis software packages that specify the number of clusters. A
number of iterations are required to find a satisfactory clustering result. The validity of the clustering result is visually assessed without applying the formal performance measures. It is difficult to measure the clustering result for multi-dimensional data sets for the users.

The k-means clustering acts as a preprocessing tool. The clusters are determined by the certain specific requirements of the main processing algorithm. A black box is employed by the k-means algorithm without validating the clustering result. The K-means clustering is also considered as a typical method for the partitioning clustering. Certain common ideas of the k-means clustering are shared depending on various hypotheses, models and criteria. The probabilistic method uses the statistical measures that are not applicable in the k-means algorithm.

A cost function of the k-means algorithm is used to determine the k-value because the cost function is modified. The data sets are integrated with the uniform distribution that is utilized to adjust and verify the clustering result. In the k-means algorithm, the benchmarking data sets are extracted from the machine learning databases. The account information of the k-means clustering reflects the performance of the k-means clustering algorithm. The clustering results of the k-means clustering algorithm are assessed without considering the performance information. By using this clustering approach, difficulty may occur while comparing the quality of the clusters. It is not used in non-globular clusters.

The mixture of Gaussians is a well-weathered model of applied statistics to provide a widespread belief in its fundamental importance. In a typical application, the sample data originates from several possible sources. Each particular source is modeled by a Gaussian. This type of distribution is common in the physical sciences and finds the theoretical justification in the central limit theorem. The main goal of the given mixed and unlabeled data from a weighted combination of these sources is to identify the generating mixture of the Gaussians. The Gaussian source is divided into mean and covariance. The mixture of Gaussians is used in a various fields such as psychology, geology and astrophysics. This mixture of Gaussians is used among widely used statistical models. The current techniques of the mixture of Gaussians are local search heuristics with weak performance guarantees, so that the Jaccard algorithm is introduced for similarity clustering.


1.8 SOFTWARE MODEL

A software model is an abstract representation of a process. It presents a specific description from the particular perspective, such as specification, design, validation and evolution. The phases of the software cycle are described by software life cycle models. The requirements are translated into design. The code is produced during the implementation process. After the implementation process, the testing process is performed in the requirements. The main purpose is used to gather the requirements of the project managers and the stakeholders. A stakeholder meeting is conducted to determine the requirements.

Figure 1.5 - Software Development Model

Figure 1.5 shows the software development model. The requirement specification and system specification are given to perform an acceptance test plan. A system integration test plan is performed in both the system specification and the system design. A sub-system integration test plan is performed in the system design and the detailed design. A sub-system integration test is performed in the sub-system integration test plan. The acceptance and system integration test are performed in the corresponding acceptance test plan and the system integration test plan to provide a service.

The software system design is produced from the results of the requirements phase. The software design is one part of deliverables of a design phase. A code is produced for implementing the software in the software development life cycle. The implementation process is overlapped with both design and testing phases. During the testing process, the
implementation is tested to solve addressing and gathering the requirement needs. The software development process contains different models as listed below:

- **Waterfall model**: It is a classical model of the software engineering. It consists of multiple non-overlapping stages. This model starts with establishing system requirements and software requirements. It serves as a baseline for multiple lifecycle models. It is easy to understand and implement. This model has a wide application range.

- **Iteration model**: The issues created in the waterfall model are overcome by this model. It requires less up-front information to provide rapid results. It needs a development team help to demonstrate results earlier in the process and obtain valuable feedback from the user.

- **V-shaped model**: It is a sequential path of execution of processes. A testing process is emphasized in this V-shaped model greater than the waterfall model. Once the coding is completed, then the testing process is begun. It provides a higher chance of success over the waterfall model due to the early development of test plans during the software development life cycle.

- **Spiral model**: It is similar to the incremental model that is integrated with several emphasis placed on risk analysis. Four phases of the spiral model are planning, risk analysis, engineering and evaluation. The requirements are gathered in the planning phase. The risk and alternate solutions are identified in the risk analysis phase. The software is produced early in the software life cycle with high amount of risk analysis.

- **Extreme model**: This model is developed based on the development and delivery of very small increments of functionality. The team members are used to characterize the agile methods. The priority changes are very difficult, when multiple stakeholders are available in the software model.

In each and every model, the requirements gathering, design, implementation, testing and maintenance processes are performed. The waterfall model and the iteration model are the two commonly used software development model. After performing the
clustering, the software model is developed to satisfy the entire requirements of the stakeholders.

1.9 SOFTWARE RISK EVALUATION

Risk is a combination of two factors, namely, the probability of malfunctioning (failure) and the consequence of malfunctioning (severity). Risk management defines the activity to identify a risk and define the policies or strategies of the risk. A software requirement’s risk is mainly used to address the possibility of loss of any functional or non-functional requirement of the software system to provide difficulty while make changes to the software system under the generation of the software model. The risk evaluation is concerned with assessing probability and impact of individual risks for the stakeholders. In general, the probability is evaluated by using the likelihood of the particular outcome of the stakeholders during risk evaluation. Thereafter, the impact is evaluated according to the result or effect of the stakeholder. The important elements are related to the impact factor such as time, quality, benefit and resource. Some of the risk available in the software model is evaluated in terms of numerical terms. The values of the probability and the impact are mainly used to evaluate the risk tolerance.

In the field of Information technology, risk management is a critical factor. The process of risk management embodies the identification, analysis, planning, tracking, controlling and communication of risk. It provides a visibility into threats to project success to provide a better structured mechanism. The risk management is primarily used to reduce the uncertainty. The risk analysis is displayed in the software design phase to evaluate the criticality of the system. The risks are analyzed and necessary counter measures are introduced. The three dimensions of the software risk are namely, technical risk, organizational risk and environmental risk. The technical risk results from uncertainty in the task and procedure. The organizational risk results from poor communication and organizational structure. The environmental dimension obtains the results from rapidly changing environment and problems between the external relationship with software developers and/or users. A Software Risk Evaluation (SRE) is a process of identifying, analyzing and developing mitigation strategy risks in a software-intensive system.
1.10 MOTIVATION

In the existing work, the severe negative consequences have occurred in the society because of the software model generator which is unaware of product design on diverse stakeholders. The high range of stakeholder’s requirement prediction is complex in this situation so that the effective and efficient requirement engineering activities are essential for the software systems to meet the expectation of the users and the customers. This research work is mainly focused on the software model development based on the stakeholder requirements. A specification of the stakeholder requirements is analyzed and clustered to identify the homogeneous group of the stakeholders.

In this research work, the preprocessed stakeholder’s requirement is taken as input. The feature evaluation, feature selection and the fuzzy classification are performed to predict the exact stakeholder’s requirements. A stakeholder’s network model is formulated depending on the Bayesian network model for the relevancy prediction of the requirements. A Jaccard similarity mechanism is applied to predict the similarity between the stakeholder’s requirements to generate a software model.

1.11 OBJECTIVE

The main objectives of the thesis work are as follows:

- To evaluate the stakeholder’s data by preprocessing the stakeholder requirements.
- To classify the stakeholder’s data based on a fuzzy classification algorithm.
- To cluster the stakeholder’s data based on Jaccard Similarity with Stake Requirement Clustering Algorithm (SRCA).
- To generate an accurate software model according to the stakeholder’s requirements with better accuracy.

1.12 SCOPE AND DELIMITATION

The purpose of this research is to evaluate the Requirement Management processes. In order to generate the software model, the stakeholder’s data is evaluated by preprocessing the stakeholder’s requirements. A hybrid approach of filter and wrapper is
used for feature evaluation and feature selection. It is inexpensive and mainly used to reduce computational complexity. To access large amount of stakeholder’s requirements during classification, the SRCA algorithm is introduced. The classified stakeholder’s data are clustered depending on the Jaccard Similarity, which evaluates the similarity among the stakeholders. Hence, an accurate software model is generated according to the stakeholder’s requirements. This software model is mainly developed to satisfy the stakeholder’s needs on time with better accuracy when compared with other existing approaches.

1.13 ORGANIZATION OF THESIS

The thesis is organized as follows:

Chapter 2 provides a review of the relevant literature regarding generation of the software model.

Chapter 3 presents proposed method to generate the software model according to the stakeholder’s requirements.

Chapter 4 provides a detailed description and comparison of the performance between the proposed method and the existing method.

Chapter 5 provides the results and discussion of the results of the research work.

Chapter 6 concludes the proposed work with key points for enhancement that can be carried out in future.