İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A FUZZY MULTI CRITERIA DECISION MAKING APPROACH TO SOFTWARE LIFE CYCLE MODEL SELECTION

M.Sc. Thesis by
Mümin HİÇDURMAZ

Department : Industrial Engineering

Programme : Industrial Engineering

JUNE 2011

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A FUZZY MULTI CRITERIA DECISION MAKING APPROACH TO SOFTWARE LIFE CYCLE MODEL SELECTION

M.Sc. Thesis by
Mümin HİÇDURMAZ
(507071117)

Date of submission : **06 May 2011**
Date of defence examination: **08 June 2011**

Supervisor (Chairman) : **Prof. Dr. Ramazan EVREN (ITU)**
Members of the Examining Committee : **Prof. Dr. Cengiz KAHRAMAN (ITU)**
Prof. Dr. Füsun ÜLENGİN (DU)

JULY 2011

<u>İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ</u>

YAZILIM YAŞAM DÖNGÜSÜ MODELİ SEÇİMİ İÇİN BİR BULANIK ÇOK KRİTERLİ KARAR VERME YAKLAŞIMI

YÜKSEK LİSANS TEZİ

Mümin HİÇDURMAZ
(507071117)

Tezin Enstitüye Verildiği Tarih : 06 Mayıs 2011
Tezin Savunulduğu Tarih : 08 Haziran 2011

Tez Danışmanı : Prof. Dr. Ramazan EVREN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Cengiz KAHRAMAN (İTÜ)
Prof. Dr. Füsun ÜLENGİN (DÜ)

TEMMUZ 2011

FOREWORD

Software is being used in a wide variety of application areas and managing software projects is a difficult task to deal with. In order to manage software projects effectively, decision making needs to be in every stage of the software development process. Using decision making techniques in software engineering management will enable to accomplish the desired goals and guarantee the satisfaction of stakeholders.

I would like to thank Prof. Dr. Ramazan EVREN for his valuable advice and to my wife due to her valuable encouragement and support, and to my colleagues for their great comments during the preparation of the thesis.

May 2011 Mümin Hiçdurmaz

(Industrial Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

AHP	: Analytic Hierarchy Process
ANP	: Analytic Network Process
BSC	: Balanced Scorecard
COTS	: Commercial of the Shelf
CR	: Consistency Ratio
DEA	: Data Envelopment Analysis
DM	: Decision Maker
FAHP	: Fuzzy Analytic Hierarchy Process
FNIS	: The Fuzzy Negative Ideal Solution
FPIS	: The Fuzzy Positive Ideal Solution
IEEE	: Institute of Electrical and Electronics Engineers
ISM	: Interpretive Structural Modeling
IT	: Information Technology
ISO	: International Organization for Standardization
MCDM	: Multi Criteria Decision Making
PROMETHEE	: Preference Ranking Organization Method for Enrichment Evaluations
SLCM	: Software Life Cycle Model
TFN	: Triangular Fuzzy Number
TOPSIS	: Technique for Order Preference by Similarity to Ideal Solution

LIST OF TABLES

Page

LIST OF FIGURES

Page

A FUZZY MULTI CRITERIA DECISION MAKING APPROACH TO SOFTWARE LIFE CYCLE MODEL SELECTION

SUMMARY

Software is in a wide variety of application areas in todays world and is essential for all kinds of businesses. Developing high quality software for business success is therefore prime importance. For ensuring software quality, software engineering project management needs to be in every stages of the life cycle.

Lack of proper and sufficient software engineering project management cause the projects to fail, to have problems with time, budget and required features. However, the establishment of effective and efficient software project management practices still remains a challenge to software organizations.

As software engineering project management needs planning, coordinating and controlling of whole development process, many decisions need to be made to guarantee the satisfaction of the stakeholders', requirements and goals, and help software engineers greatly to implement products or applications. In brief, decision making is an essential process that must be used in the software development process.

In software engineering project management, one of the critical issues is the selection of the appropriate SLCM, which may affect the success of the project. All the stages of software development process is established due to the model selected, so SLCM selection is sufficient for enabling all the effort be used efficiently in all phases of the project life cycle.

A fuzzy multi criteria decision making approach is proposed in the study, since fuzzy sets are inevitable in representing uncertainty, vagueness and human subjectivity. Fuzzy numbers are used for representing linguistic or uncertain data. Moreover, fuzzy AHP and fuzzy TOPSIS are used together in the proposed approach for obtaining reliable results and reaching the result with logical and easy calculations. An application is done using the proposed method and a conclusion is given at the end of the study.

YAZILIM YAŞAM DÖNGÜSÜ MODELİ SEÇİMİ İÇİN BİR BULANIK ÇOK KRİTERLİ KARAR VERME YAKLAŞIMI

ÖZET

Yazılım, bugünün dünyasında çok geniş bir uygulama alanına sahip ve her türlü iş için bir gereksinim konumundadır. Dolayısıyla, yüksek kalitede yazılım üretmek her türlü iş başarısı için vazgeçilmez bir öneme sahiptir. Yazılım kalitesini sağlamak için, yazılım mühendisliği proje yönetimi yazılım yaşam döngüsünün her aşamasında yer almalıdır.

Yazılım mühendisliği proje yönetiminin olmaması yada yeterli olmaması, projelerin zaman, bütçe ve gerekli özellikleri yerine getirememekten dolayı başarısız olmalarına sebep olmaktadır. Diğer yandan, etkin ve verimli yazılım projesi yönetimi halen, yazılım organizasyonları için bir zorluk olarak karşımıza çıkmaktadır.

Yazılım mühendisliği proje yönetimi planlama, koordinasyon ve geliştirme aşamalarının kontrolünü gerektirdiğinden, paydaşların memnuniyetini, gereksinimleri ve hedefleri garantileyecek ve yazılım mühendislerine ürün ve geliştirmelerin uygulanmasında önemli kolaylık sağlayacak kararlar verilmelidir. Özetle, karar verme, yazılım geliştirme sürecinde uygulanması gereken süreçlerden biri olmalıdır.

Yazılım mühendisliği proje yönetimindeki kritik konulardan birisi de, projenin başarısını etkileyebilecek öneme sahip olan yazılım yaşam döngüsü modeli seçimidir. Yazılım geliştirme sürecinin tamamı seçilen model üzerine kurulduğundan, yazılım yaşam döngüsü modelinin seçimi projenin tüm aşamalarında işgücünün verimli bir şekilde kullanılması açısından vazgeçilmez bir unsurdur.

Bulanık kümeler, belirsizliği,kararsızlığı ve insan subjektifliğini temsil etmede en etkin metodlardan birisi olduğundan, bu çalışmada bir bulanık çok kriterli karar verme yaklaşımı önerilmiştir. Bulanık sayılar dilsel ve kesin olmayan verilerin temsilinde kullanılmıştır. Ayrıca, önerilen yaklaşımda, bulanık AHP ve bulanık TOPSIS metodlarının birlikte kullanılması, güvenilir sonuçlar elde etmek ve sonuca mantıklı ve kolay hesaplanabilir bir yoldan gitmek için tercih edilmiştir. Önerilen yaklaşım kullanılarak bir uygulama yapılmıştır. Çalışmanın son bölümünde ise sonuç bölümüne yer verilmiştir.

1. INTRODUCTION

1.1 Software Project Management Problems

Software is being used in an increasingly wide variety of application areas, and is often critical for business success nowadays. The software industry is entering a period of maturity, while at the same time software is becoming a crucial component of many of today's products. Developing high quality software products is therefore of prime importance. There exists two approaches for ensuring product quality, one being assurance of the process by which the product is developed, and the other being the evaluation of the quality of the end product.

Many improvement methods on software engineering project management are carried out for managing software development process in order to produce high quality products. However, many software projects still have problems to deliver on time, within budget, with all the required features and functions. Besides, the establishment of effective and efficient project management practices still remains a challenge to software organizations.

According to the Standish Group's CHAOS Summary 2009 Report, only 32% of all projects were delivered on time and on budget, with required features and functions. However, 44% were challenged which are late, over budget, and/or with less than the required features and functions. Moreover, 24% failed which are cancelled prior to completion or delivered and never used [1].

Among the reasons for those problems is a lack of project management. Several problems occur in software project development process due to the lack of project management. According to a survey by Emam and Koru (2008), the reasons for project cancellations and failures include:

- Requirements and scope changes
- Lack of necessary management skills
- Over budget

- Lack of necessary technical skills
- No more need for the system to be developed
- Over schedule
- Too new technology
- Insufficient staff
- Critical quality problems with software
- Insufficient involvement of senior management and end users [2].

Moreover, there exists several other factors that lead to software project failures like, organizational structure, unrealistic or unarticulated goals, use of wrong software development methodologies, poor reporting of the project status, unmanaged risks, undefined processes, commercial pressures, poor leadership and personality conflicts [3].

Most of the reasons mentioned above that cause software projects failure are managerial ones. Hence, effective management of software engineering projects has become increasingly important to the success of both government and commercial enterprises. Software project managers require methods to plan, monitor, and control the complex software processes and products.

1.2 Definitions

We should start by defining the terms that will be used during the study.

Project management: Project management is the application of knowledge, skills, tools, and techniques to project activities in order to meet project requirements [4].

Software engineering management: The application of management activities like planning, coordinating, measuring, monitoring, controlling and reporting to ensure that the development and maintenance of software is systematic, disciplined and quantified [5].

Software life cyle model: Framework containing the processes, activities and tasks involved in the development, operation and maintenance of a software product, spanning the life of the system from the definiton of its requirement to the termination of its use [6].

As software engineering management needs planning, coordinating and controlling of whole development process, many decisions need to be made to guarantee the satisfaction of the stakeholders', to meet the requirements and goals, and to help software engineers greatly to implement products and applications. In brief, decision making is an essential process that must be used in the software development process.

In software engineering management, one of the critical issues is the selection of the appropriate software life cycle model (SLCM) which may affect the success of the project. The adaptation and deployment of appropriate software life cycle model must be done in light of the scope, magnitude, complexity and requirements of the project [6]. Selection of the right SLCM helps to decompose the project into tasks, with associated inputs, outputs, and completion.

The aim of SLCM selection is to enable all the effort be used efficiently in all phases of the project. SLCM selection can be considered as evaluating the specific needs and challenges of a project and then choosing the most appropriate model for the software development process. The main benefit of model selection is enabling the development efficiency by ensuring the tasks ordering that are well suited to the needs of a specific project [7]. Although choosing the right life cycle model has no inherent risks, the model selected may contain additional risks that can result in missing tasks and inappropriate task ordering and may cause to project failure.

1.3 Purpose of The Thesis

The purpose of this study is to propose a new fuzzy multi criteria decision making approach to software life cycle model selection. There exists no systematic approach or study about the selection of SLCM by using MCDM methods. That is why, this study fulfills the need for the use of MCDM methods in SLCM selection. It is often necessary to consider many factors related with people, process, technology and etc. in SLCM selection. Hence, MCDM methods are useful to solve this kind of problems that have a large set of criteria to consider. Moreover, in order to corporate with qualitative and quantitative criteria, fuzzy set theory is used in the proposed approach.

This study provides a wide view of existing software life cycle models, important factors to be considered in SLCM selection, and a new approach for SLCM selection. Section 2 gives a review of decision making and MCDM studies in software engineering field. Section 3 gives information about existing SLCMs. In section 4, a literature review about the factors that have to be considered in selecting SLCM is given. Section 5 gives basics of the fuzzy sets theory and the MCDM methods that is used in the proposed approach. In section 6, a new fuzzy MCDM approach to SLCM selection is proposed and an application is given. Moreover, a sensitivity analysis is done in order to analyze the results gained. The offer for future studies and a conclusion is given at the end of the thesis.

2. SOFTWARE ENGINEERING DECISIONS

2.1 Decision Making in Software Development Process

In software development process, many choices need to be made and decisions to be taken in order to guarantee the satisfaction of the stakeholders. A stakeholder can be virtually anyone, that has something to do with the project, like end users, project team members, senior management and even sub contractors.

The critical issues that needs decisions to be taken can be determining of the non-functional requirements and the order of the implementation of these, selection of appropriate architecture design style or combination of styles that best satisfies a set of quality attributes, choosing the right software life cycle model, choosing the best tools that will be used in software development process, and etc.

2.2 Related Studies

Many research works exist in the literature about software engineering decisions by several authors. Ahmad and Laplante present a rigorous model for selecting a software project management tool using the Analytical Hierarchy Process (AHP) [8]. AHP is chosen as it can be understood easily by the decision maker and be implemented with a flexible, systematic, and repeatable evaluation procedure. In addition, this work establishes a framework for comparing individual product decisions across projects, project managers, organizational groups, and organizations.

In another work by the same authors , a framework for operating system selection is developed with AHP. By explicitly representing preference, providing tools that allow users to set and inspect their judgements, and affording users with systematic evaluation procedure, the contribution of this study is to help the decision maker to better identify an appropriate real-time operating systems solution without the need for intensive performance testing [9].

As an extension of use of AHP, Tamura and Yamada propose a reliability assessment method based on the AHP [10]. Moreover, AHP is used to assess the quality of ensemble methods in software defect prediction by Yi et al. [11] . On the other hand, Trienekens et al. uses AHP in the work which defines an approach for software developers to improve the way that they deal with software quality [12]. Syamsuddin and Hwang introduce a framework to guide decision makers evaluating information security policy performance. The framework which adopts AHP methodology, is developed into a four level hierarchy (goal, criteria, sub-criteria, and alternatives) representing different aspects of information security policy [13].

AHP is applied in several works in combination with different decision making methods. For instance, Rajesh proposes an effective decision making framework for software selection using a multiple criteria decision making method, Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE). The method is improved by integrating it with AHP and fuzzy logic. Fuzzy logic is introduced to handle the imprecision of the human decision making process [14]. Kanungo and Monga present a prioritization scheme based on the AHP to obtain individual and aggregate ranks of process improvement ideas as a part of software process improvement in an organization. Moreover, they have shown how complementarities between combinations of process change requests can be identified by integrating AHP and Interpretive Structural Modeling (ISM) [15].

Fuzzy logic and AHP is used for software development strategy selection in an another study. The study is based on the extent fuzzy AHP modeling to deal with the uncertainty and vagueness from subjective perception and experience of humans in the decision process [16].

Fuzzy AHP and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is also used by Ballı and Korukoğlu to select appropriate operating system for computer systems of the firms by taking subjective judgments of decision makers [17]. Fuzzy Vikor and Fuzzy Delphi is combined in a study for measuring the performance of software development projects [18]. Thomaidis et al. present a fuzzy set-based approach to the evaluation of information technology projects [19].

Four MCDM methods, TOPSIS, PROMETHEE, Data Envelopment Analysis (DEA) and ELECTRE are examined together for determination of user preferences based software defect detection algorithms selection[20].

Table 2. 1: Summary of MCDM studies on software engineering management

Author	Year	Aim/Subject	Method
Yi et al.	2011	Software defect prediction	AHP
Asosheh et al.	2010	Evaluation of information technology projects	DEA BSC
Peng et al.	2010	Determination of user preferences based software defect detection algorithms selection	DAE TOPSIS ELECTRE PROMETHEE
Syamsuddin and Hwang	2010	Security policy decision making	AHP
Trienekens et al.	2010	Specification, prioritization and metrication of software product quality	AHP
Ahmad and Laplante	2009	Commercial real-time operating systems selection	AHP
Ballı and Korukoğlu	2009	Operating system selection	Fuzzy AHP TOPSIS
Rajesh	2009	Software selection in manufacturing industries	AHP PROMETHEE Fuzzy Logic
Büyüközkan and Ruan	2008	Evaluation of software development projects	Fuzzy Vikor Fuzzy Delphi
Ahmad and Laplante	2006	Software project management tool selection	AHP
Shyur	2006	COTS evaluation	ANP Modified TOPSIS
Tamura and Yamada	2006	Software reliability assessment	AHP
Thomaidis et al.	2006	Evaluation of information technology projects	Fuzzy Sets
Kanungo and Monga	2005	Prioritization of software process change requests	AHP and ISM
Büyüközkan et al.	2004	Software development strategy selection	Fuzzy AHP

In addition, modified TOPSIS method is considered in a work which aims to model the Commercial of the Shelf (COTS) evaluation problem as MCDM problem. A five-phase COTS selection model, combining the technique of Analytic Network Process (ANP) and modified TOPSIS is proposed [21]. Moreover, Balanced Scorecard (BSC) and DEA is compared in another wok for IT project selection [22]. MCDM studies on software engineering management are given in Table 2. 1.

3. SOFTWARE LIFE CYCLE MODELS

3.1 Software Life Cycle Model Selection

All the activities and work products necessary to develop a software system constitutes a software life cycle model (SLCM). According to IEEE 12207, a SLCM is a framework containing the processes, activities and tasks involved in the development, operation and maintenance of a software product, spanning the life of the system from the definition of its requirement to the termination of its use [6].

SLCM describes the major phases of development and define the major processes and activities. SLCM also specifies products of each of the phases and inputs at the beginning of the phases and provides a framework for the activities to be mapped [7]. SLCM aids managers and developers to deal with the complex process of developing software. There exists several life cycle models in order to understand, measure and control the software development process better [6].

The IEEE 12207 standart does not require the use of any particular SLCM, but it does require each project to define a suitable SLCM[6].

Software life cycle process management is consisting of four phases as given below [6]:

- Select the appropriate SLCM to deliver and support the products
- Create the software life cycle by identifying and defining tasks
- Establish the software life cycle process
- Manage the software life cycle process throughout the products' identified life

So it is essential to select a SLCM before creating the software life cycle and establishing the process. The IEEE 1074 standart gives the steps to be followed to select a SLCM as given below [23]:

- Identify all the SLCMs available to the development organization

- Identify the attributes that apply to the desired end system and the development environment
- Identify any constraints that may be imposed on the selection
- Evaluate the various SLCMs using lessons learned in past projects
- Select the SLCM that will best satisfy the steps above

According to the steps listed above, firstly the available SLCMs need to be identifed.

3.2 Software Life Cycle Models

There exists several SLCMs in the literature. Each of the models has advantages and disadvantages, and there is no specific rule that one model is best for all kind of projects.

3.2.1 Waterfall model

Waterfall is the first published model of the software development process that was derived from the sytem engineering process by Royce in 1970 [24]. The model is an activity-centered classical model of development software that is usually called the conventional model. The waterfall model is a sequential SLCM, in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing (validation), integration and maintenance [25].

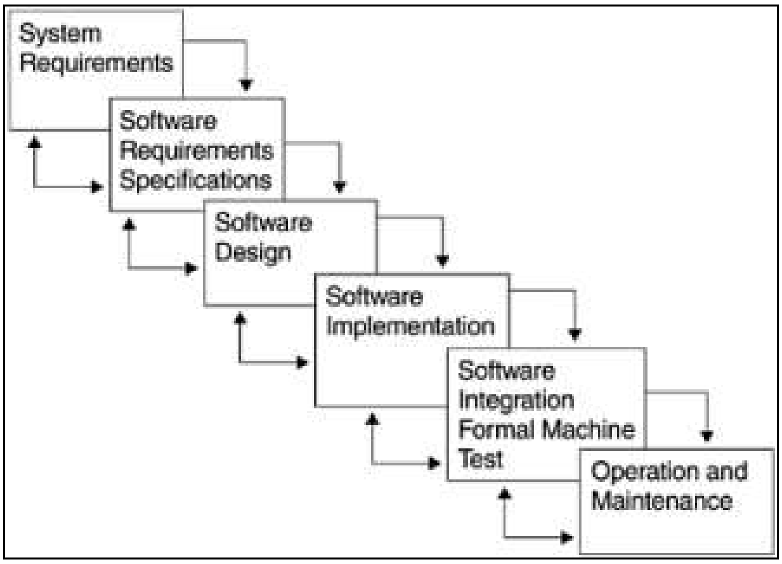

Figure 3.1 : Waterfall model [26]

This model progress down the path through each of the phases with deliverables like software requirements specification, design documents, code and etc. at each stage [6]. Each phase consists of a definite set of activities and deliverables that must be accomplished before the following phase can begin.

The key point in this model is to never turn back once an activity is completed. So it is essential to follow each activity and phase by a review. In brief, the model assumes that software development process can be scheduled as a step-by-step process that transforms user needs into code [27].

The advantages of this model are [6, 26, 27]:

- Enables early specification of the system and its structure
- Enables more accurate tracking of project progress, early identification of possible slippages and measurable software development
- Projects more manageable and delivered on time without cost overrun
- Ease of predicting budget and effort
- Reviews at the end of each stage ensure user involvement
- Generates documents used to test and maintain the system
- Conserve resources with minimizing wasted effort
- Works well for technically weak or inexperienced staff

The disadvantages of this model are [6, 24, 26, 27]:

- Customers must express the requirements completely, correctly and with clarity
- Too much time spent on planning and documentation
- An extensive effort for integration and test is required at the end of the project
- No demonstration is available before the end of the project
- Changes in requirements and backing up to address mistakes is difficult and costly
- Lack of flexibility
- Hard to predict all needs in advance

- Design flaws not discovered until the testing phase

The structural approach of this model makes it suitable for large organizations with large and complex development projects.

3.2.2 V model

V-Shaped or V Model is a variation of the waterfall model that has a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing is emphasized in this model more than the waterfall model. The testing procedures are developed early in the life cycle before any coding is done, during each of the phases proceeding implementation. Requirements begin the life cycle model just like the waterfall model. Before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in requirements gathering. The high-level design phase focuses on system architecture and design. An integration test plan is created in this phase in order to test the pieces of the software systems ability to work together. However, the low-level design phase lies where the actual software components are designed, and unit tests are created in this phase as well. The implementation phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V, where the test plans developed earlier are now put to use [6, 28].

The advantages of this model are [28, 29]:

- Simple and easy to use
- Each phase has specific deliverables
- Higher chance of success over the waterfall model due to the early development of test plans during the life cycle
- Works well for small projects where requirements are easily understood

The disadvantages of this model are [28, 29]:

- Very rigid like the waterfall model
- Little flexibility and adjusting scope is difficult and expensive
- Software is developed during the implementation phase, so no early prototypes of the software are produced

- Does not provide a clear path for problems found during testing phases.

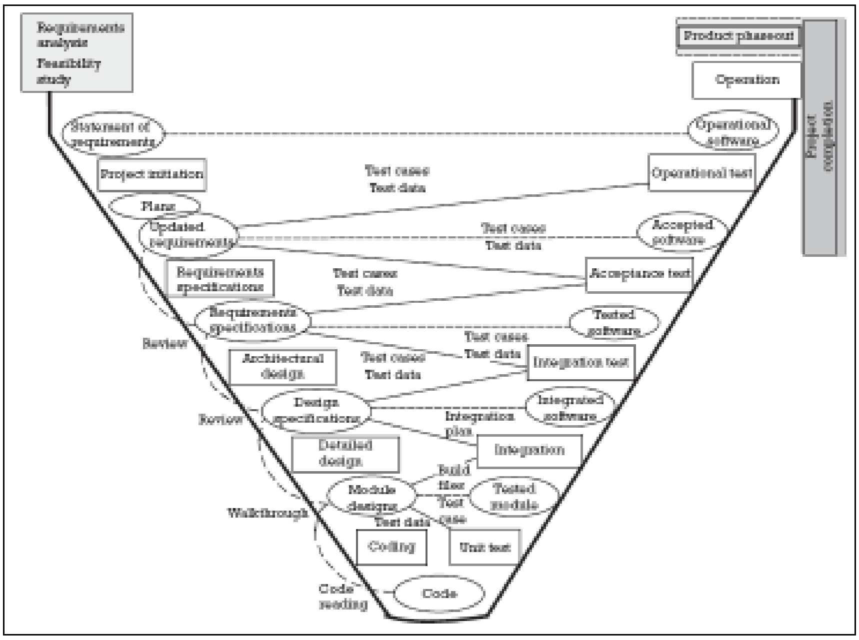

Figure 3.2 : V model [29]

3.2.3 Spiral model

The Spiral Model is developed by Barry Boehm in 1988. In the Spiral Model, development effort is iterative and, that as soon as one iteration is completed, another iteration commences [6]. The model is an activity-centered model aim to address the source of weaknesses in the waterfall model. Its main goal is to accommodate the infrequent changes during the software development. The model has risk management, reuse and prototyping activities in addition to same activities that waterfall model has. The extended activities are done in cycles and rounds. Each round follows the waterfall model and includes determining objectives, specifying constraints, generating alternatives, identfying and resolving risks, developing and verifying next level product and planning activities [27].

Another feature of the spiral model is that, only one cycle of the process may actually develop software deliverables. Starting at the center of the spiral, one can see that each development phase (concept of operation, software requirements, product design, detailed design, and implementation) involves one cycle of the spiral [26].

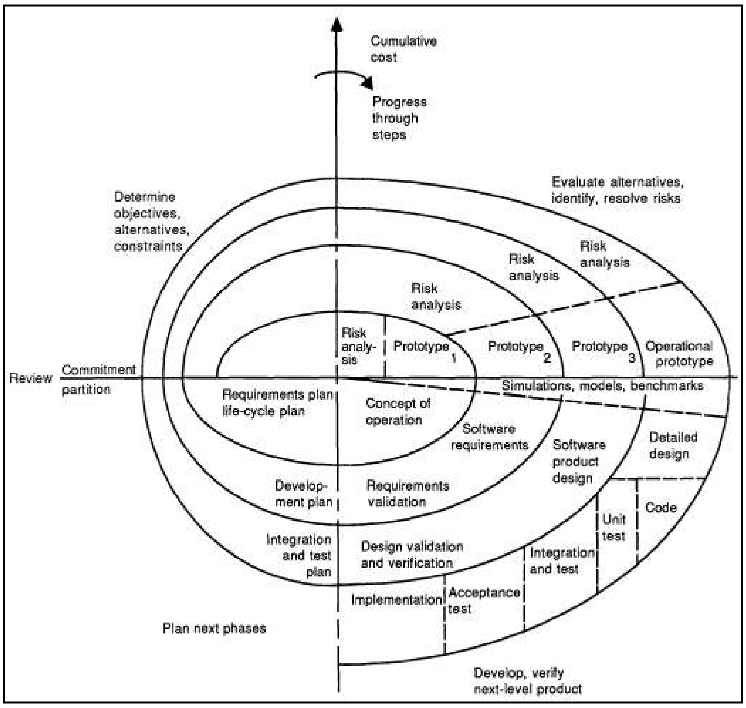

Figure 3.3 : Spiral model [30]

The model is good when dealing with high risk development projects and with a client who is exactly not sure of the requirements like real time applications. The model is used mostly for large government projects [30].

The advantages can be listed as given below [30]:

- Range of options accommodates the good features of existing models
- Risk-driven approach avoids many of their difficulties
- Accommodates preparation for life cycle evolution, growth and changes of the software product
- Incorporates software quality objectives into software product development
- Focuses on eliminating errors and unattractive alternatives early

However, the model has several difficulties and disadvantages given below [30]:

- Not determining specific deadlines may end up waterfall model like
- The flexibility and freedom may cause losing accountability and control for contract software
- Need for further elaboration of spiral steps so that consistency, tracking and control can be achieved

3.2.4 Incremental model

The Incremental Model is again a variation of waterfall model which has iterations. The model can be used when the requirements can be segmented into an incremental series of products that are developed independently. At the beginning, the project is divided into small parts. This allows the development team to demonstrate results earlier in the process and obtain valuable feedback from system users. Often, each iteration is actually a mini-Waterfall process with the feedback from one phase providing vital information for the design of the next phase. In a variation of this model, the software products, which are produced at the end of each step (or series of steps) can go into production immediately as incremental releases [28].

Moderate control is maintained over the life of the project through the use of written documentation, formal review and approval by the user and technology management at designated major milestones. Stakeholders can be given concrete evidence of project status throughout the life cycle [25].

The model is useful when requirements are well known at the initial phase and the product can be divided into independent deliverables called build increments [6]. Communication and coordination skills take central stage in project development. Moreover, it enables knowledge sharing as the knowledge gained at the design of the first increment can be transfered to the design of the second increment.

The advantages of incremental model are [6, 25]:

- Less cost and time is required to make the first delivery
- Provide faster results, require less up-front information and offer greater flexibility
- Smaller system development enable less risk
- Incremental funding is allowed
- Customer involves all stages and quick to implement

The disadvantages are [6, 25, 28, 31]:

- Increments might be withdrawn from service, reworked and rereleased if requirements are not stable or complete

- Difficult implementation issues delayed
- Some modules will be completed much earlier than others
- Well-defined interfaces are required
- User feedback following each phase may lead to increased customer demands

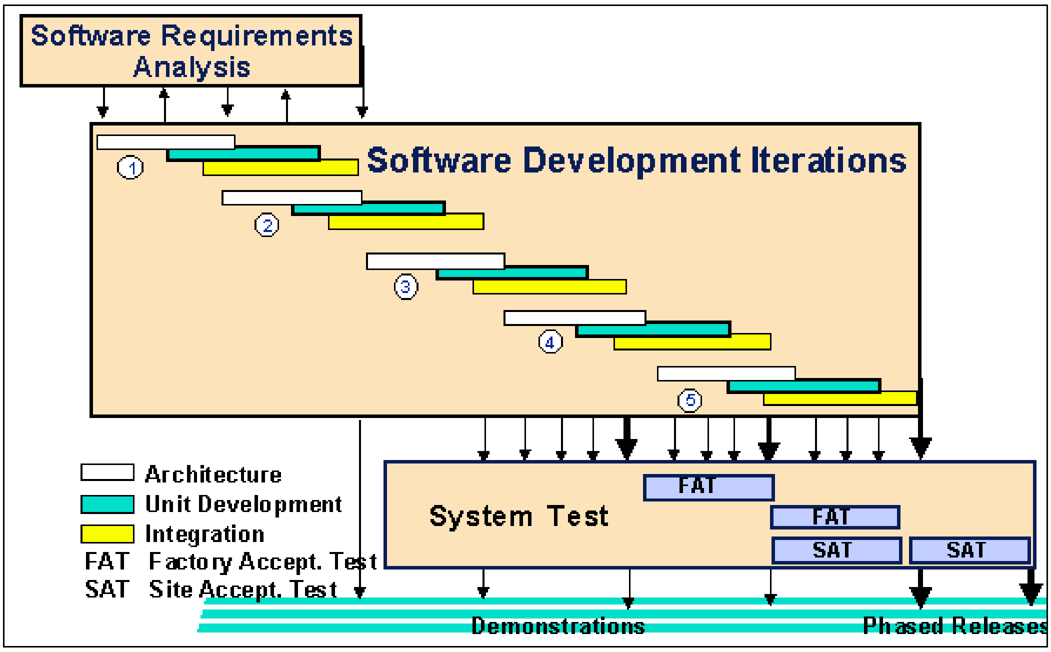

Figure 3.4 : Incremental model [31]

3.2.5 Evolutionary prototyping model

The Evalutionary Prototyping model is based on the idea of developing an initial implementation, offer this to user comment and refining it through many versions until the adequate system has been developed [24]. The model explicitly extends the incremental model to the requirements phase. The first build increment is used to refine the requirements for a second build increment. The first increment to users is released and this provide feedback that will assist in the development of requirements for the later increments. Moreover, developing a build increment will provide visibility into issues that were not recognized prior to actually starting work on that increment. Once the requirements are understood the phases of design, coding can be implemented by waterfall model within incremental development model [6, 26].

The main advantage of this model is its having the ability to address risk early in the project, early feedback on whether the final system will be acceptance and visible progress throught the project. The factors that are important in using this model is using experienced developers, managing schedule and budget expectations and managing the prototyping activity itself [24].

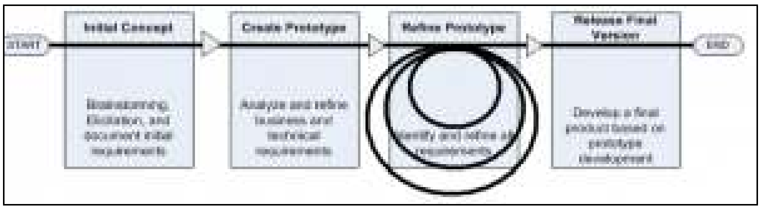

Figure 3.5 : Evolutionary prototyping model [26]

It is probably best suited to business systems in which developers can have frequent, informal interactions with end-users. This is useful when requirements are changing rapidly, when the customer is reluctant to commit to a set of requirements, or when no one fully understands the application area. However, some risks will occur in using the model. The main risks associated with model are unrealistic schedule and budget expectations, inefficient use of prototyping, unrealistic system performance and poor design [24].

The problems that will be encountered using the model can be classified as management, maintenance and contractual. Existing management processes assume a waterfall model of development and specialist skills are required which may not be available in all development teams are management problems. Furthermore, continual change tends to corrupt system structure so long-term maintenance is expensive [26].

3.2.6 Unified model

Unified model is another life cycle model similar to Boehm’s Spiral Model in which a project consists of several cycles, each of which ends with the delivery of a product to the customer. Each cycle consists of four phases which are inception, elaboration, construction and transition. Again each phase consists of a number of iterations.

In the inception phase, an idea is defined and its feasibility is evaluated. In the elaboration phase, the project is planned, the system is defined and resources are allocated. The construction phase corresponds to the development process while transition phase corresponds to the installation and post-development process. Moreover, the Unified Model assumes that requirements, analysis, design, implementation and testing participate in each of these iterations which emphasize the staging of resources, an aspect of software development that is not captured in other SLCMs [27, 32].

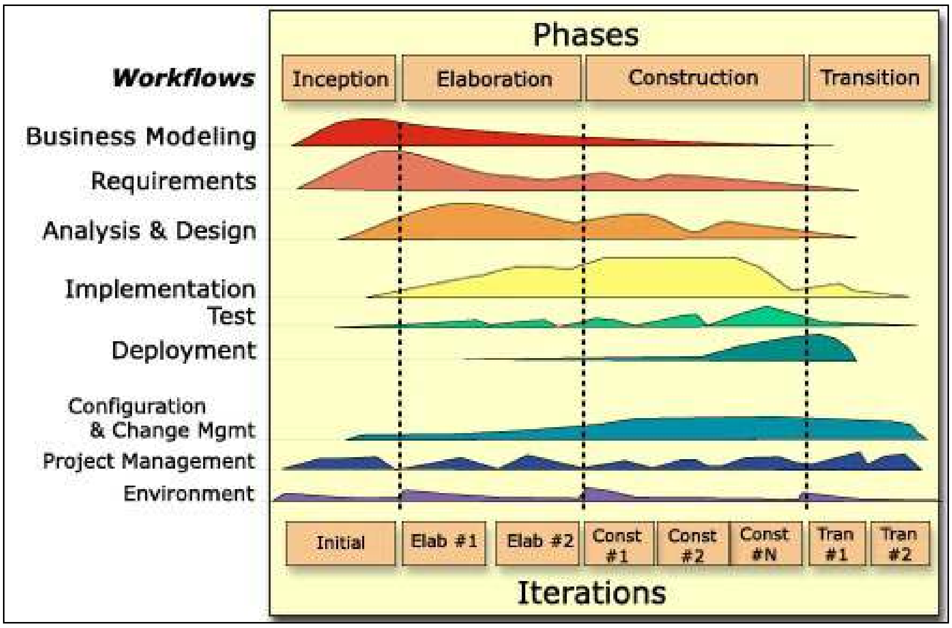

Figure 3.6 : Unified model [32]

High tracebility of the model allows understand the effect of changes. Using a component based architecture creates a system that is easily extensible, promotes software reuse and intuitively understandable. The model allow less technically competent individuals who may have a better understanding of the problem to have a greater input. Besides, managing requirements using use-cases and scenarios have been found to be very effective at both capturing functional requirements and help in keeping sight of the anticipated behaviors of the system. Iterative and incremental design helps reduce project risk profile, allows greater customer feedback and help developers stay focused [27, 32].

4. IMPORTANT FACTORS IN SOFTWARE LIFE CYCLE MODEL SELECTION

It is difficult to compare and contrast SLCMs as all of them have different characteristics. Moreover, there exists common attributes for all of them that must be considered in the selection of most appropriate SLCM for the specific project. Although the literature does not give any systematic study about the use of MCDM methods in SCLM selection, important factors in SLCM selection are considered in many other studies, books and standarts.

Due to IEEE 12207, project scope, magnitude, complexity, changing needs are the most important factors that play important role in SLCM selection [6]. The degree of experience in application domain, type of the project and complexity are the other critical factors that mentioned in another study [33]. Christensen and Thayer lists the factors like the tolerance of the model to the risks, requirements known degree, importance of early (partial) functionality, complexity, requirements stability, maturity of the application, availability and priority of funding, flexibility and criticality and importance of processes and documentation [27].

Hanafiah and Kasirun propose a model for selecting the right life cyle model for software development. They state that some assessments have to be done by experts. From the characteristics of the models they agree that size,complexity, requirement stability, duration, criticality, modularity, process and documentation requirement, user interface requirement, risk assessment, project team and sufficiency of resources are the factors that have impact on SLCM selection [34].

Alexander and Davis offer a set of criteria for SLCM selection. The set of criteria fall into five different categories which are personnel, problem, product, resource and organizational. The personel criteria is consisting of both developers and users that contain users experince in application domain, users ability to express requirements, developers experince in application domain and developers software experience. The problem criteria has maturity of the application, problem complexity, requirement for partial functionality, frequency of changes and magnitude of changes as sub-criteria.

The product criteria is consisting of product size, product complexity, non-behavioral requirements and human interface requirements. The resource criteria is consisting of funding profile, funds availability, staffing profile, staff availability and accessability of users. The last criteria contains management capability and quality assurance and configuration management capability [35].

In another study, a set of criteria is used to define the differences between SLCMs. The set is consisting of need of intensive planning, documentation and quality control, the need for formal review, flexibility, knowledge of the team, type of the product developed, risk management, discovery of errors, feedback, delivery speed, user involvement, communication and coordination, customer demand rate, emphasize of testing, focus on system architecture and design, project size, customer evaluation of products, amount of cost, type of project or critically of the project, code improvement rate/continous improvement, quality of the design and skill need [28].

Another work give a set of criteria that are derived from different comparison view points of life cyle model. The examples of the criteria set are flexibility, ease of management, requirements stability, software risk management, project size, criticality, project's priorities, the need for value and quality and people [36].

Moreover, innovation patterns, organizational learning and knowledge management are the other critical elements to be considered. Adaptable and flexible ways of working is gaining importance in todays rapidly changing enviroments and technology. The project manager must select the appropriate SLCM for the project that will be developed taking into the changing factors and the characteristics of the project developed [36].

Sharma and Gupta extracted ten project main risks namely, personnel, schedule, process, functionality, safety, user or client involvement, performance, reliability, financial and maintainability from the literature. These risks have to be considered in SLCM selection acoording to the authors. A four-level hierarchical model for software project success is established. The objectives of budget performance, schedule performance and quality performance that contribute to the goal occupy the second level of the hierarchy. The ten main project risk-related factors take place in the third level of the hierarchy and can be considered in three different contexts (budget, schedule and quality). The ten main project risks occupy the immediate lower level. Each of the sub-risk factors occupy the lowest level of the hierarchy, corresponding to one of those ten project risks [37]. The project, process and technical criteria considered as important in SLCM selection by several authors are given in the Table 4.1. The criteria related with people are given in Table 4.2.

Table 4.1: Literature survey on project, process and technical criteria

Criteria	ISO 12207 [6]	Bruegge and Dutoit [33]	Christensen and Thayer [27]	Hanafiah and Kasirun[34]	Davis et al. [44]	Kettunen and Laanti [36]	Sharma and Gupta [37]
Scope	X						
Size	X			X	X	X	
Complexity	X	X	X	X	X	X	
Requirements' stability	X		X	X	X	X	
Requirements known degree			X				
Early delivery			X		X		
Maturity of the application			X		X	X	
The availability of the funding			X		X	X	X
Flexibility			X			X	
Criticality			X	X		X	
Planning, process and documentation			X	X		X	X
Modularity, adaptability				X		X	
Sufficiency of resources				X	X	X	X
Human interface requirements				X	X		
Quality assurance and configuration management capability					X	X	
Formal review need		X					
Integration and testing						X	

Table 4.1 and Table 4.2 give us too many criteria that are considered as important in SLCM selection. However, some of the criteria that are considered by different authors may have the same meaning. Moreover, the criteria considered may not have be in the same level of importance. For example, some of the criteria considered can be sub-criteria of the others.

Table 4.2: Literature survey on criteria related with people

Criteria	ISO 12207 [6]	Bruegge and Dutoit [33]	Christensen and Thayer [27]	Hanafiah and Kasirun [34]	Davis et al. [44]	Kettunen and Laanti [36]	Sharma and Gupta [37]
Team experince in application domain		X		X	X		
Team experince				X	X		X
Risk affect			X	X		X	
Users experience in application domain					X		
Users ability to express requirements					X		
User involvement and feedback						X	X
Communication and coordination						X	
Management capability					X	X	

Scope is a factor that has been considered, but it is a general term related with cost, budget and resources. That is why, it is not meaningful to take scope as a criterion in the decision model. Size is a factor that has a wide variety of meaning again related with cost, budget, duration and resources. Complexity, flexibility, criticality, modularity and adaptability can be considered as process sub-criteria and they can be thought under the process criterion. Requirements' stability and requirements known degree are the factors that can be combined and considered as requirements management. Users ability to express requirements is an important factor that must be taken into account in SLCM selection, as feedback is important in all engineering disciplines. Team experince in software engineering and application domain are the factors mentioned in the literature, however they do not directly affect the selection of SLCM.

Risk affect directly affects the SLCM selection, as all the alternative models have different approaches for software risk management. In software development process risks are generally related with stakeholders, so risk affect or mainly risk management can be consired under the people criterion. Communication and coordination and management capability are the factors directly related with people and can be considered under the people criterion.

Early delivery and testing and integration are the technical factors. Early delivery is the result of testing and integration, so it can be regarded under testing and integration criterion.

Maturity of the application factor have the similar meaning with requirements known degree. Because, if the maturity of the application is high, the requirements known degree is again high. Maturity of the application factor can not be considered as a seperate criterion and can be inside the requirements known degree criterion.

Planning, process and documentation is an important factor and can be a sub-criterion of process criterion. Quality assurance and configuration management capability are again can be regarded as process sub-criteria. The availability of funding and sufficiency of resources are directly related with management and cost criteria.

Formal review need and human interface requirements are the other technical factors that must be taken into account.

5. FUZZY SETS AND MCDM METHODS

5.1 Fuzzy Sets Theory

Zadeh originally describes fuzzy as fuzzy set, which is a technique that is designed to cope with imprecise linguistic concepts or fuzzy terms. It allows users to provide inputs in imprecise terms and receive either fuzzy or precise advice [38].

A linguistic variable is a variable whose values are words or sentences in a natural or artificial language. For instance, some matters are characterized by linguistic term in nature, such as good, medium and bad. Each linguistic variable may be assigned one or more linguistic values, which are in turn connected to a numeric value through the mechanism of membership functions [39].

Some basic definitions and notations of fuzzy sets and fuzzy numbers are reviewed from the literature and will be presented [38, 39, 40, 41].

Definition 1: Let X be a universe of discourse corresponding to an object whose current status is fuzzy, and the status value is characterized by a fuzzy set $\overline{A}$ in X. A membership function $\mu_{\overline{A}}(x): X \rightarrow [0,1]$ is called the membership function of $\overline{A}$. It connects with each element x in X, a real number in the interval [0,1]. The function value $\mu_A(x)$ is termed the grade of membership of x in $\overline{A}$.

$$\mu_{\overline{A}}(x) = \begin{cases} 0, \text{ where } x < l \\ (x-l)/(m-l), \text{ where } l \le x \le m, \\ (u-x)/(u-m), \text{ where } m \le x \le u, \\ 0, \text{ where } x > u. \end{cases} \qquad \textbf{(5. 1)}$$

Definition 2: A triangular fuzzy number (TFN) can be defined by a triplet (l, m, u), where u is greater than m and m is greater than l. Mathematical form of a triangular fuzzy is displayed in the following equation.

Definition 3: A fuzzy number $\overline{A}$ is a normal and convex fuzzy subset of X, which is described as $\sup_x \mu_{\overline{A}}(x) = 1$,

$$\mu_{\overline{A}}(x)[\lambda x_1 + (1-\lambda)x_2] \geq \min[\mu_{\overline{A}}(x_1), \mu_{\overline{A}}(x_2)] \quad \textbf{(5. 2)}$$

Because of the definition of fuzzy number as $\overline{A} = (l, m, u)$, arithmetic operations of fuzzy numbers depends on the arithmetic operations on the interval. Some main operations for fuzzy numbers decsribed are as follows:

Definition 4: Let $\widetilde{a} = (l_1, m_1, u_1)$ and $\widetilde{b} = (l_2, m_2, u_2)$. Then the addition is defined as the following;

$$\widetilde{a} + \widetilde{b} = (l_1, m_1, u_1) + (l_2, m_2, u_2) \quad = (l_1 + l_2, m_1 + m_2, u_1 + u_2). \quad \textbf{(5. 3)}$$

Subtraction is

$$\widetilde{a} - \widetilde{b} = (l_1, m_1, u_1) - (l_2, m_2, u_2) \quad = (l_1 - l_2, m_1 - m_2, u_1 - u_2). \quad \textbf{(5. 4)}$$

Multiplication is

$$\widetilde{a} \times \widetilde{b} = (l_1, m_1, u_1) \times (l_2, m_2, u_2) \quad = (l_1 \times l_2, m_1 \times m_2, u_1 \times u_2) \quad \textbf{(5. 5)}$$

$$k\widetilde{a} = k \times (l_1, m_1, u_1) \quad = (k \times l_1, k \times m_1, k \times u_1). \quad \textbf{(5. 6)}$$

Division is

$$\widetilde{a} \div \widetilde{b} = (l_1, m_1, u_1) \div (l_2, m_2, u_2) \quad = (l_1 \div l_2, m_1 \div m_2, u_1 \div u_2). \quad \textbf{(5. 7)}$$

Inverse is

$$\widetilde{a}^{-1} = (l_1, m_1, u_1)^{-1} = (\frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1}). \quad \textbf{(5. 8)}$$

Definition 5: Let $\widetilde{a} = (l_1, m_1, u_1)$ and $\widetilde{b} = (l_2, m_2, u_2)$ be two triangular fuzzy numbers, then the vertex method is defined to calculate the distance between them, as the following:

$$d(\widetilde{a}, \widetilde{b}) = \sqrt{\frac{1}{3}\left[(l_1 - l_2)^2 + (m_1 - m_2)^2 + (u_1 - u_2)^2\right]}. \tag{5. 9}$$

5.2 Multiple Criteria Decision Making

Multiple criteria decision making (MCDM) is an effective tool widely used for evaluating and ranking problems. These problems are usually complex engineering problems that have incomplete and vague information. The MCDM approach enables the choice to be made among decision alternatives described by their attributes [42 ,43]. In the next section, MCDM methods AHP and TOPSIS, extended in a fuzzy environment will be presented.

5.3 Fuzzy AHP

One of the MCDM method is AHP which is extensively used for modelling unstructured problems in many fields such as economics, social, and management science [45].

In order to deal with the uncertainity and vagueness from the subjective perception in decision-making process, many fuzzy AHP methods are proposed by various authors. AHP is used with fuzzy logic as fuzzy logic provides a simple way to reason with vague, ambiguous, and imprecise input, and decision makers usually find it more confident to give internal judgements than fixed value judgements [45].

Fuzzy AHP methods are systematic approaches to the alternative selection by using the concepts of fuzzy set theory and hierarchical structure analysis [38]. Van Laarhoven and Pedrcyz proposed the first studies that applied fuzzy logic principle to AHP [46]. They compared fuzzy ratios described by triangular membership functions in their work [46]. Moreover, Buckley initiated trapezoidal fuzzy numbers to express the decision maker's evaluation on alternatives with respect to each criterion [47]. Chang introduced a new approach for handling FAHP, with the use of triangular fuzzy numbers for pairwise comparison scale of FAHP, and the use of the extent analysis method for the synthetic extent values of the pairwise comparison [48].

The FAHP using synthetic extent values is used to compare catering firms [49], to evaluate machine tool alternatives [50], to compare quality consultants [51] and for software development strategy selection [16].

In this part, some terms that will be used in Chang's extent analysis will be detailed [49, 51, 53].

By using TFNs via pairwise comparison, the fuzzy judgement matrix $A = (a_{ij})$ can be expressed mathematically as:

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1(n-1)} & a_{1n} \\ a_{21} & 1 & \cdots & a_{2(n-1)} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & 1 & a_{(n-1)n} \\ a_{n1} & a_{n2} & \cdots & a_{n(n-1)} & 1 \end{bmatrix}_{n \times n} \tag{5. 10}$$

The judgement matrix A is an $n \times n$ fuzzy matrix containing fuzzy numbers a_{ij}.

$$a_{ij} = \begin{cases} 1, \; i = j \\ 1,3,5 \; or \ldots \; 1^{-1}, 3^{-1}, 5^{-1}, \; i \neq j \end{cases} \tag{5. 11}$$

Let X be an object set, whereas $U = \{u_1, u_2, ..., u_m\}$ is a goal set. According to fuzzy extent analysis, the method can be performed with respect to each object for each corresponding goal g_i, resulting in m extent analysis values for each object given as $M_{g_i}^1, M_{g_i}^2, ..., M_{g_i}^m, \; i = 1, ..., n,$ where all the $M_{g_i}^j, \; j = 1, ..., m$ are TFNs representing the performance of the object x_i with regard to each goal u_j. The steps of Chang's extent analysis [52] can be detailed as follows [49, 51]:

Step 1: The fuzzy synthetic extent value with respect to the *i th* object is defined as:

$$S_i = \sum_{j=1}^{m} M_{g_i}^{j} \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} \right]^{-1} \tag{5. 12}$$

To obtain $\sum_{j=1}^{m} M_{g_i}^{j}$, perform the fuzzy addition operator m extent analysis values for a particular matrix such that

$$\sum_{j=1}^{m} M_{g_i}^{j} = \left(\sum_{j=1}^{m} l_j, \sum_{j=1}^{m} m_j, \sum_{j=1}^{m} u_j \right) \tag{5. 13}$$

and obtain $\left[\sum_{j=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} \right]^{-1}$, perform the fuzzy addition operation of $M_{g_i}^{j}\ (j = 1, 2, ..., m)$ values such that

$$\left[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} \right] = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} l_{ij}, \sum_{i=1}^{n} \sum_{j=1}^{m} m_{ij}, \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ij} \right) \tag{5. 14}$$

and then compute the inverse of the vector such that

$$\left[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} \right]^{-1} = \left(\frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{m} u_i}, \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{m} m_i}, \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{m} l_i} \right) \tag{5. 15}$$

Step 2: The degree of possibility of $M_2 \geq M_1$ is defined as:

$$V(M_2 \geq M_1) = \sup_{y \geq x} \left[\min(\mu_{M_1}(x), \mu_{M_2}(y)) \right] \tag{5. 16}$$

and can be equivalently expressed as follows:

$$V(M_2 \geq M_1) = hgt(M_1 \cap M_2) = \mu_{M_2}(d) \tag{5. 17}$$

$$= \begin{cases} 1, & if\ m_2 \geq m_1, \\ 0, & if\ l_1 \geq u_2, \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & otherwise, \end{cases} \tag{5. 18}$$

where d is the ordinate of the highest intersection point D between μ_{M_1} and μ_{M_2}.

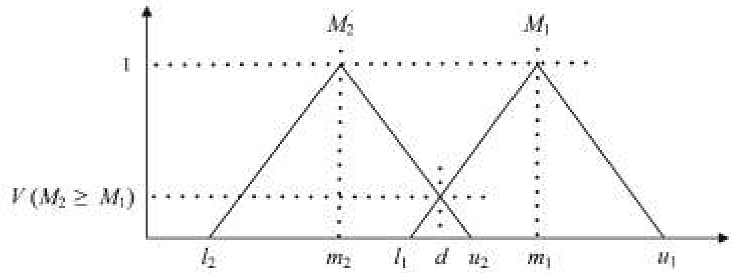

Figure 5.1: Intersection point "*d*" between two fuzzy numbers M_1 and M_2.

To compare M_1 and M_2, both the values of $V(M_2 \geq M_1)$ and $V(M_1 \geq M_2)$ are required.

Step 3: The degree possibility of a convex fuzzy number to be greater than k convex fuzzy numbers M_i $(i = 1, 2,..., k)$ can be defined by:

$$V(M \geq M_1, M_2,..., M_k)$$

$$= V\left[(M \geq M_1) \text{ and } (M \geq M_2) \text{ and},..., \text{and } (M \geq M_k)\right] \quad \textbf{(5. 19)}$$

$$= \min V(M \geq M_i) \; i = 1, 2, \ldots, k.$$

Assume that:

$$d'(A_i) = \min V(S_i \geq S_k) \quad \textbf{(5. 20)}$$

for $k = 1, 2,..., n;\; k \neq i.$ Then, the weight vector is given by:

$$W' = (d'(A_1), d'(A_2), \ldots, d'(A_n))^T \quad \textbf{(5. 21)}$$

where A_i $(i = 1, 2,..., n)$ has n elements.

Step 4: The normalized weight vectors are defined as:

$$W = (d(A_1), d(A_2), \ldots, d(A_n))^T \quad \textbf{(5. 22)}$$

where W is a nonfuzzy number.

5.4 Fuzzy TOPSIS

The technique for order preference by similarity to ideal solution (TOPSIS) is another well-known MCDM technique in which the chosen alternative should be as close to the ideal solution as possible and as far from the negative ideal solution as possible. The basic principle of this method is to find a solution with the shortest distance from the positive-ideal solution and the biggest distance from the negative-ideal solution. It uses the Euclidean distance (or any other) to calculate the shortest distance. This method is easy to understand and it ensures that the tradeoff among attributes is compensatory [54]. Besides, TOPSIS is a widely accepted multi criteria decision making technique due to its sound logic, simultaneous consideration of the ideal and the anti-ideal solutions and easily programmable computation procedure [56].

However, TOPSIS is often criticized for its inability to handle uncertainity and imprecision, as the method uses crisp values for personel judgments. So, TOPSIS is extended in a fuzzy enviroment where criteria values are represented by fuzzy numbers [55]. Also linguistic preferences can be easily converted to fuzzy numbers and TOPSIS allows using these fuzzy numbers in the calculation.

Fuzzy TOPSIS steps can be outlined as follows [56, 57]:

Step 1: Choose the linguistic ratings $\left(\tilde{x}_{ij},\ i = 1,2,3,\cdots,n,\ j = 1,2,3,\cdots,J\right)$ for alternatives with respect to criteria. The fuzzy linguistic rating ($\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij})$) preserves the property that the ranges of normalized TFNs belong to [0,1]; thus, there is no need for normalization. The linear scale transformation can be used to transform te various criteria scales into a comparable scale. Therefore, we can obtain the normalized fuzzy decision matrix denoted by $\tilde{R}$.

$$\tilde{R} = [\tilde{r}_{ij}]_{m\times n} \tag{5. 23}$$

and

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*}\right)$$

(5. 24)

$$c_j^* = \max_i c_{ij}.$$

(5. 25)

Step 2: Calculate the weighted normalized fuzzy decision matrix. The values of weighted normalized matrix are $\tilde{v}_{ij}$, which are calculated by

$\tilde{v}_{ij} = \tilde{r}_{ij} \times w_i$,

where w_i are weights of criteria respectively.

Step 3: Identify positive ideal (A^+) and negative ideal (A^-) solutions. The fuzzy positive ideal solution ($FPIS$, A^*) and the fuzzy negative ideal solution ($FNIS$, A^-) are shown in the following equation:

$$A^* = \{\tilde{v}_1^*, \ldots, \tilde{v}_i^*\}$$

$$= \left\{\left(\max_j v_{ij} | i \in I'\right), \left(\min_j v_{ij} | i \in I''\right)\right\}$$

(5. 26)

$$i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, J$$

$$A^- = \{\tilde{v}_1^-, \ldots, \tilde{v}_i^-\}$$

$$= \left\{\left(\min_j v_{ij} | i \in I'\right), \left(\max_j v_{ij} | i \in I''\right)\right\}$$

(5. 27)

$$i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, J$$

where I' is associated with benefit criteria and I'' is associated with cost criteria.

Step 4: Calculate the distance of each alternative from A^* and A^- using equations **(5.26)** and **(5.27)** as given below:

$$D_j^* = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_i^*) \; j = 1,2,\ldots,J$$

(5. 28)

$$D_j^- = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}_i^-) \; j = 1,2,\ldots,J$$

Step 5: Calculate the similarities to ideal solution.

$$CC^*{}_j = \frac{D_j^-}{D_j^* + D_j^-}, \; j = 1,2,...,J \qquad \textbf{(5. 29)}$$

Step 6: Rank preference order. Choose an alternative with maximum CC_j^* or rank alternatives according to CC_j^* in descending order.

5.5 Use of Fuzzy AHP and Fuzzy TOPSIS Together

In some kind of problems using of a hybrid MCDM method can be effective and useful. Both AHP and TOPSIS has many advantages and disadvantages, and it is not best to use one of the methods in every situation. So, it is better to use these two methods in a problem by implementing each of them in different stages of the problem. Since AHP enables to get reliable results of pairwise comparisons and TOPSIS is one of the most useful methods of ranking the best alternative, a combination of these two methods makes the decision making problem easy to deal with [43].

There exists many works with models that use fuzzy AHP and fuzzy TOPSIS. A model for evaluating government websites based on fuzzy AHP and fuzzy TOPSIS is proposed [58]. Transshipment site selection using the AHP and TOPSIS approaches under fuzzy environment is done in another work [59]. A model for machine tool selection by using fuzzy AHP and fuzzy TOPSIS is another work proposed [56]. In addition, fuzzy AHP and fuzzy TOPSIS is used together as a model in weapon selection, prioritizing effective factors in production systems, performance evaluation and selection of the strategic alliance partner in logistics value chain [57, 60, 61, 62]

6. A NEW FUZZY MCDM APPROACH TO SLCM SELECTION

6.1 Proposed Approach

This section focuses on a new fuzzy multi criteria decision making approach to software life cycle model selection. There exists no systematic study about the use of MCDM methods and even fuzzy sets theory in SLCM selection. The proposed approach fulfills the need for the use of MCDM methods and fuzzy sets theory in SLCM selection.

There exist several important factors in SLCM selection, that is why MCDM methods is useful in dealing with this kind of problems. Besides, it is often difficult to express precise statements in the evaluation process. So, fuzzy sets theory is used in order to corporate imprecise statements. The proposed approach is strong as it compares the criteria with Chang’s fuzzy AHP method which gives reliable results. Moreover, the approach includes TOPSIS for determining the alternatives’ priority weights, since it is rational and understandable and all computations can be done easily.

ANP is not used, as the sub-criteria under each criterion is not directly related with other sub-criteria under an another criterion. As an example, the cost sub-criterion under people criterion does not have a direct relationship with the flexibility sub-criterion under the process criterion. Moreover, use of ANP can be hard to implement since there can be a huge amount of pairwise comparison. That is why AHP is used as it is easy to establish a hierarchy.

In the first phase of proposed approach, alternative SLCMs are determined based on the literature survey and expert opinion. Next, the criteria and sub-criteria that will be used in the evaluation process is also determined by literature survey and expert opinion. A decision hierarchy is constructed using the criteria, sub-criteria and alternatives determined. A decision making team is formed and a detailed questionnaire is conducted for the evaluation procedure at the last step of the first phase.

In the second phase, fuzzy AHP is used for assigning the criteria and sub-criteria weights by using the results of questionnaires.

In the last phase, fuzzy TOPSIS is used to determine alternatives' priority weights. Rank of alternatives is obtained and the best SLCM is offered at the end. The schematic diagram of the proposed approach is given in Figure 6.1.

6.1.1 Determining alternative SLCMs

In the first step of the proposed approach, the alternative SLCMs are determined. In the determination of the alternatives, both literature survey and expert opinion are used. The fundamental books, standarts and journals are investigated. Software engineering methods and tools are changing rapidly, so the popularity of the models are taken into account in selecting alternative models.

6.1.2 Determining the criteria to be used in the evaluation

In the second step of the proposed approach, a detailed review is done in order to specify the important criteria in SLCM selection. Both literature survey and expert opinion are used for specifying the criteria set for the evaluation. The fundamental books, standarts and journals are also investigated.

It is necessary to consider many factors related with people, process, technology and etc. in SLCM selection. The important criteria for SLCM selection are also given in Section 4.

6.1.3 Structuring decision hierarchy

Using the alternatives and the criteria set determined, a decision hierarchy is established. Literature survey on decision making problems is useful for establishing the decision hierarchy.

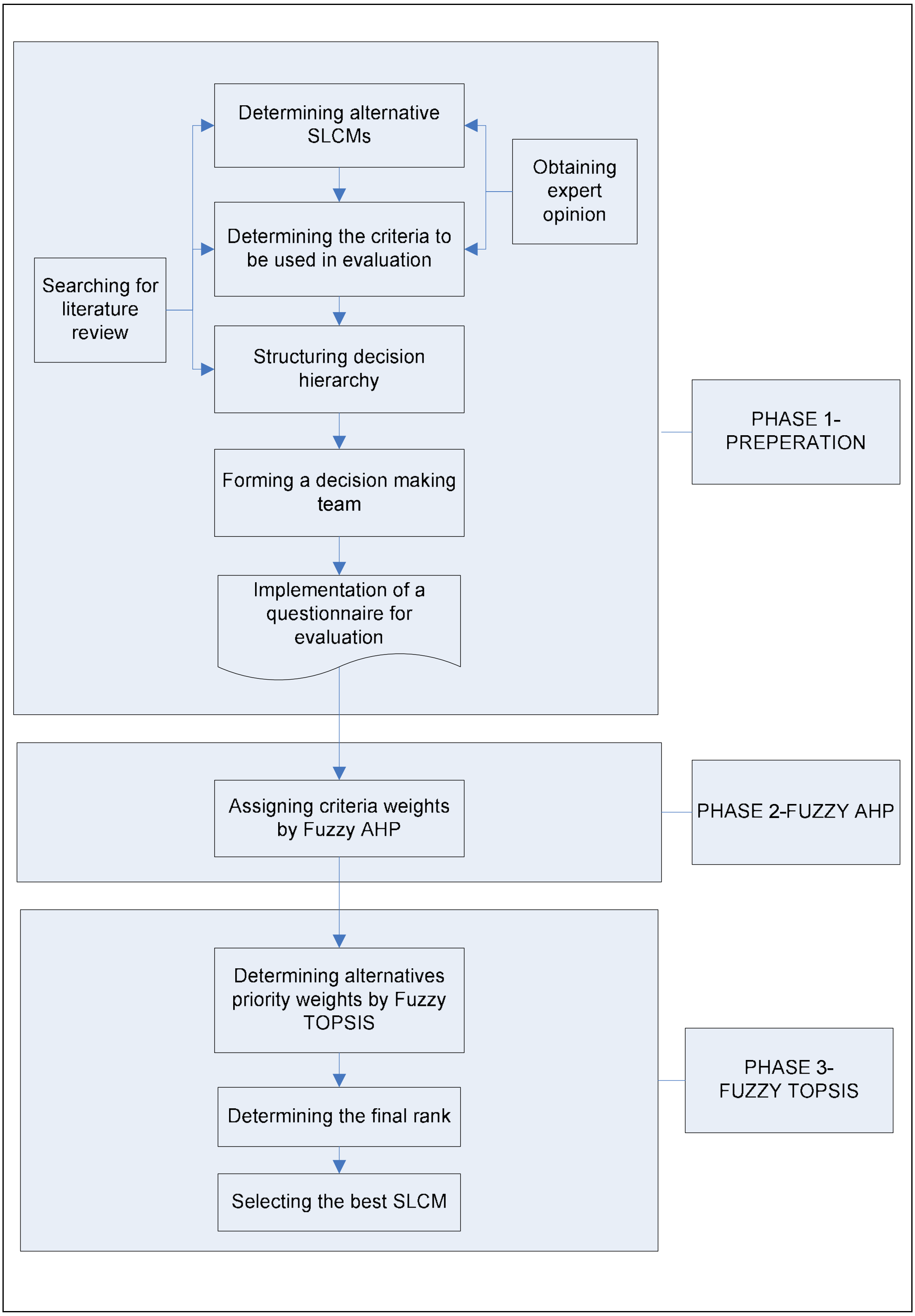

Figure 6.1: Schematic diagram of the proposed approach

6.1.4 Forming a decision making team

A decision making team is formed for assessment process. Team members need to have sufficient experience and knowledge in software engineering [17]. The weights of the decision makers in the evaluation process are also determined.

6.1.5 Implementation of a questionnaire for evaluation

A detailed questionnaire is prepared for the evaluation procedure. A brief information about the decision makers takes place in the first part of questionnaire. In the second part, comparison of each criterion and also each sub-criterion is presented. In the last part, evaluation of each alternative with respect to criterion or sub-criterion is presented to decision makers.

6.1.6 Assigning criteria weights by fuzzy AHP

Triangular fuzzy numbers are used in assigning criteria and sub-criteria weights. Using an appropriate method, evaluations of decision makers are aggregated.

Fuzzy AHP is used to determine the relative weights of evaluation criteria. If the problem has sub-criteria, then fuzzy AHP is also used to determine the relative weights of sub-criteria. AHP gives reliable results as it allows pairwise comparisons [60].

6.1.7 Determining alternatives' priority weights by fuzzy TOPSIS

Fuzzy TOPSIS is used to determine alternatives' priority weights. Before the computations, the benefit and cost criteria have to be determined [56]. TOPSIS is one of the most useful methods for ranking the best alternative and it enables eliminating many procedures to be performed [43]. Besides, TOPSIS is rational, easily understandable and can be calculated easily [60].

Triangular fuzzy numbers are used in determining alternatives priority weights. Using an appropriate method, evaluations of decision makers are aggregated.

6.1.8 Determining the final rank

Final rank of alternatives is decided according to alternatives' relative closeness to ideal solution.

6.1.9 Selecting the best SLCM

The first ranked SLCM is selected as the best alternative for the decision making problem at the last step.

6.2 Application

In this section, an application is presented for the selection of SLCM for a selected project in an organization using the new proposed approach. Firstly, the alternatives and criteria are determined based on the literature survey and expert opinion. A four-level decision hierarchy is established in the next step. A decision making team is formed and a detailed questionnaire is conducted for the evaluation procedure.

In the second phase, using the results obtained, fuzzy AHP is used for assigning the criteria and sub-criteria weights. In the third phase, fuzzy TOPSIS is used for determining alternatives' priority weights. The final rank is determined and the best SLCM for the selected project is offered.

6.2.1 Determining the alternatives

Based on the literature survey and experts opinion, four SLCMs are determined as alternatives. These models are Waterfall, V Model, Spiral and Evalutionary Prototyping. One of the reasons for selecting these four models is their popularity nowadays. Moreover, they are fundamental and commonly used models in software engineering.

For example, Throwaway Prototyping is eliminated because the experts indicated that it was used before 2000 and it is not used nowadays. Similarly, Iterative Model is eliminated due to new models like Spiral and V model which are derived from Iterative Model.

6.2.2 Determining the criteria

Fundamental books on software engineering and electronic databases Science Direct, IEEE and ACM are investigated to determine the criteria and sub-criteria that will be used for evaluation. The criteria set given in Section 4 is considered in the determination process. Moreover, expert opinions are also obtained from the project in the organization where the application is done.

Table 6.1: Criteria to be used in evaluation of alternatives

Criteria	Subcriteria	Explanation
People	Ease of Management	The capability of getting people together to accomplish desired goals and objectives through the software project life cycle that comprise planning, organizing, leading and controlling activities. Predictability, visibility, risk management, communication and coordination are included in this criterion
	User Involvement and Feedback	The participation of the users by evaluating, commenting, rejecting, or approving the product during its development in order to develop a product that meets users' needs
Process	Cost	Cost related with staff , training, tools, and etc. that will occur during the software development life cycle
	Complexity	The degree of difficulty to understand, build and verify of the design or implementation of a process
	Criticality	The degree of impact that a requirement, module, error, fault, failure, or other item has on the development process
	Flexibility	The ease of modification of a system or process for use in applications or environments other than those for which it was specifically designed
	Reusability	The degree of usability of a component, module or any part of the system that was developed in previous development stages in further process
Technical	Documentation and software quality	Documentation are plans, product documents, and the quality is the degree of fullfilment of the customer needs and expectations with the developed software product
	Testing and integration	Combining parts, modules or components together in order to enable to work together in a system, and trying to find any non-conformance in the product developed before it reaches to the end user
	Focus on design and architecture	The degree of emphasis or importance of software design and architecture used for software development process
	Requirements management	The management of customers needs and requirements, adaption of the changing needs to the software environment
	Formal reviews	The control of the document, component or anything that is developed in determined stages (for example, requirements review, design review, document review, and etc.)

Some of the criteria are eliminated due to expert opinions while some of them are combined as they indicate common or similar meanings. Besides, some criteria like "reusability" and "focus on design and architecture" which does not exist in Section 4 are included based on the expert opinion.

Literature survey and expert opinion give us three main criteria: people, process and technical. Moreover, these three criteria has sub-criteria.The people criterion has ease of management, user involvement and feedback and cost as sub-criteria. The process criterion has complexity, criticality, flexibility, reusability and documentation and software quality as sub-criteria. Technical criterion has requirements management, testing and integration, focus on design and architecture and formal reviews as sub-criteria.

Before the evaluation procedure, each determined criteria and sub-criteria is clearly defined in order to give a common understanding to decision makers. The definitions are given in Table 6.1.

6.2.3 Decision hierarchy of the problem

A four-level hierarchical model which is proposed in Figure 6.2 is established for the SLCM selection process since the problem has four alternatives, three criteria and 12 sub-criteria. The first level of the hierarchy indicates the goal which is selecting the best SLCM. The second level includes criteria and the third level includes sub-criteria determined. At the fourth level, the alternatives are shown.

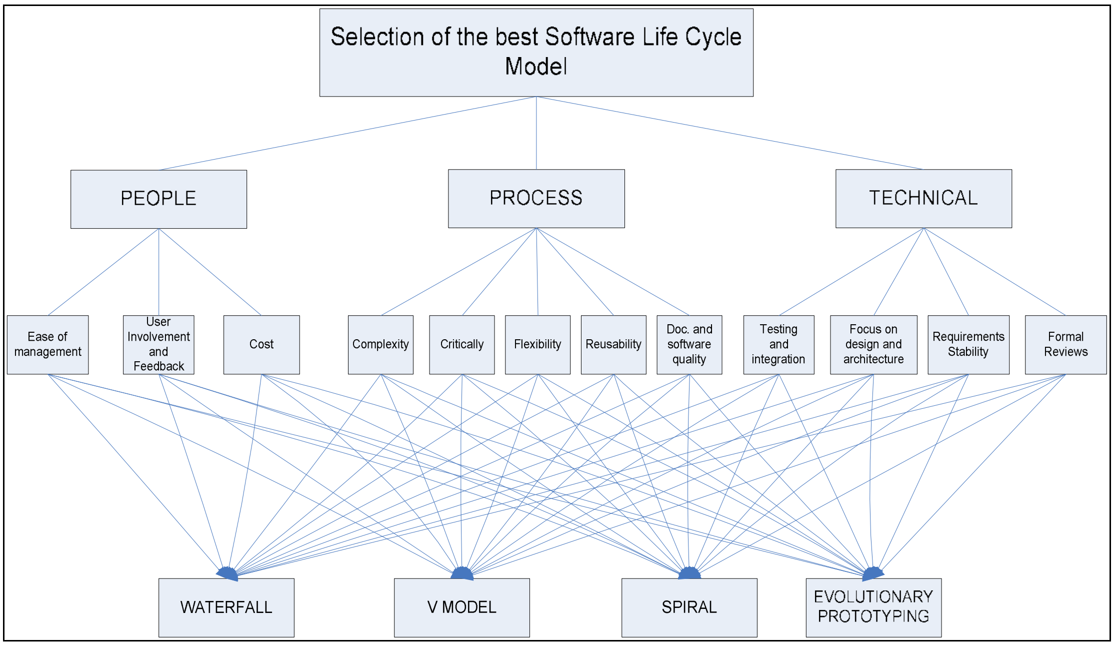

Figure 6.2: Four level hierarchical model for SLCM selection

6.2.4 Decision making team

A committee of four decision makers (DM1, DM2, DM3 and DM4) is formed to assess and select the most suitable SLCM. The team members are the project manager, the team leaders(2) and software architect. These members are chosen due to their experience for several years in both software and application domain. Besides, all of them have at least a masters degree in software engineering. This ensures that the decision makers have sufficient knowledge and experience and will give dependable answers.

In the study, it is assumed that degrees of the importance for four DMs are equal.

6.2.5 Questionnaire application

A detailed questionnaire is prepared. The first part includes brief information on the subject and aim of the study. It also includes the way decision makers will follow in filling in the questionnaire. Also brief information about decision makers will be obtained in this part. In the second part, comparison of each criterion with respect to goal, and the comparison of each sub-criterion with respect to criterion will be done by decision makers using linguistic variables. In the last part, evaluation of each alternative with respect to each sub-criterion by using linguistic variables will be done. The questionnaire is given in appendices.

6.2.6 Assigning criteria and sub-criteria weights

In this study, the linguistic variables that are utilized in the second stage (fuzzy AHP) for pairwise comparisons can be expressed in positive TFNs for each criterion as in Figure 6.3 [56].

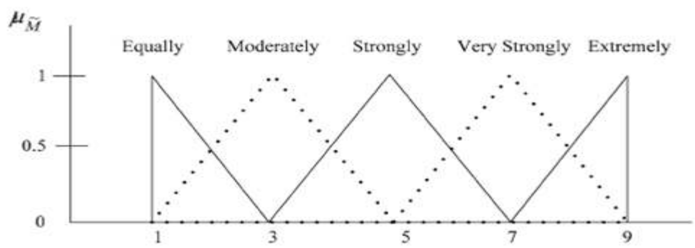

Figure 6.3: Linguistic variables for the weight of each criterion

The linguistic variables matching TFNs are presented in Table 6.2. The DMs will utilize the following linguistic weighting set to evaluate the importance of the SLCM criteria and sub-criteria: Just equal, equal importance, weak importance, strong importance, very strong importance and extremely preferred. For the associated fuzzy numbers, Table 6.2 presents the fuzzy AHP comparison scale considering the linguistic variables that describe the importance of criteria. In this phase, the decision makers are given the task of forming individual pairwise comparison matrix by using the scale given in Table 6.2. For example, if someone considers that the criterion i has "strong importance" over the criterion j, then s/he sets $a_{ij}=(3.0, 5.0, 7.0)$. So, the criterion j is thought to have "strongly less important" over the criterion i. Then, the comparison between j and i can be found using the equation 5.8 which is $a_{ji}= (1/u, 1/m, 1/l)=(1/7, 1/5, 1/3)$.

Table 6.2: Linguistic variables

Linguistic variables used for criteria weight determination	
Just equal	(1.0, 1.0, 1.0)
Equal importance	(1.0, 1.0, 3.0)
Weak importance	(1.0, 3.0, 5.0)
Strong importance	(3.0, 5.0, 7.0)
Very strong importance	(5.0, 7.0, 9.0)
Extremely preferred	(7.0, 9.0, 9.0)
If factor i has one of the above numbers assigned to it when compared to factor j, then j has the reciprocal value compared with i.	Reciprocal=$(1/u, 1/m, 1/l)$

Firstly, the importance weights of criteria with respect to goal and sub-criteria with respect to criteria are evaluated. This procedure is illustrated by the comparison results of DM 1 on criteria in Figure 6.4 and Table 6 4. Some abbreviations for criteria and sub-criteria are that will be used in order to track the study easily are presented in Table 6.3.

DM 1 indicates that, "people" has strong importance over "process", "technical" has weak importance over "people" and "technical" has weak importance over "process". This linguistic variables are returned into crisp values which have lower, medium and upper values.

This procedure is used in other decision makers' comparisons on every criterion and sub-criterion. Firstly, the importance weights of criteria with respect to goal and sub-criteria with respect to criteria are evaluated. The linguistic variables are taken from the DMs, and then returned into crisp values which have lower, medium and upper values. Geometric mean operations are used for aggregating group decisions because they are used commonly within the application of the AHP [64]. From the aggregated values of 4 DMs, pairwise comparison matrices are obtained.

Table 6.3: Corresponding abbreviations of criteria and sub-criteria

Criteria	Abb. for Criteria	Sub-criteria	Abb. for Sub-criteria
PEOPLE	(C_1)	Ease of management	C_{11}
		User involvement and feedback	C_{12}
		Cost	C_{13}
PROCESS	(C_2)	Complexity	C_{21}
		Criticality	C_{22}
		Flexibility	C_{23}
		Reusability	C_{24}
		Documentation and software quality	C_{25}
TECHNICAL	(C_3)	Testing and integration	C_{31}
		Focus on design and architecture	C_{32}
		Requirements management	C_{33}
		Formal reviews	C_{34}

With respect to goal: Software life cycle model selection												
Criteria	**Importance of one criteria over another**											**Criteria**
	Extremely preferred	Very strong importance	Strong importance	Weak importance	Equal importance	Just equal	Equal importance	Weak importance	Strong importance	Very strong importance	Extremely preferred	
People			X									Process
People								X				Technical
Process								X				Technical

Figure 6.4: A sample for comparison in the questionnaire of DM1

Table 6 4: Assignment of the sample comparison of DM1 with TFNs

	DM1	
C_1	(3.00, 5.00, 7.00)	C_2
C_1	(0.20, 0.33, 1.00)	C_3
C_2	(0.20, 0.33, 1.00)	C_3

Consistency of obtained pairwise comparison matrices need to be analyzed in the next step. In fuzzy AHP, the elements of the comparison matrix have lower, medium and upper values. That is why the first step to make consistency test in fuzzy AHP is defuzzification of the elements of comparison matrix. Kwong and Bai (2003) propose the following approach for defuzzification of triangular fuzzy numbers.

Let $M = (l,m,u)$ be a triangular fuzzy number, then $M_d= (l+4.m+u)/6$ is the crisp value of the given TFN.

Using the steps of consistency analysis [66], the CR values are obtained as 0.10, 0.09, 0.04 and 0.10 for matrices of criteria comparison and sub-criteria comparisons with respect to people, process and technical criteria respectively. Since CR is less than 0.10 for all comparison matrices, the results are acceptable and can be used.

Also, by applying equation (5.12), the fuzzy synthetic values (S_{Ci} , where C_i are criteria) are computed and then, they are used to obtain V values. V values are calculated by using equations (5.17) and (5.18). Firstly, an example computation is shown.

We consider the first comparison (criteria with respect to goal), then the fuzzy synthetic extent values of each criterion can be obtained as:

S_{C1}=(2.115, 2.985, 4.350) ⊗ (1/15.456, 1/9.716, 1/7.370)=(0.137, 0.307, 0.590)

S_{C2}=(1.623, 2.268, 3.076) ⊗ (1/15.456, 1/9.716, 1/7.370)=(0.105, 0.233, 0.417)

S_{C3}=(3.632, 4.463, 8.030) ⊗ (1/15.456, 1/9.716, 1/7.370)=(0.235, 0.459, 1.090)

Using these vectors and equations (5.17) and (5.18),

$V(S_{C1} \geq S_{C2})=1.000$

$V(S_{C1} \geq S_{C3})=0.700$

$V(S_{C2} \geq S_{C1})=0.791$

$V(S_{C2} \geq S_{C3})=0.446$

$V(S_{C3} \geq S_{C1})=1.000$

$V(S_{C3} \geq S_{C2})=1.000$

are obtained. Then, using equation (5.19), we get

$V(S_{C1} \geq S_{C2}, S_{C3})$=0,700, $V(S_{C2} \geq S_{C1}, S_{C3})$=0,446, $V(S_{C3} \geq S_{C1}, S_{C2})$=1,000.

The pairwise comparison matrix and V values are presented in the tables Table 6.5 to Table 6.8.

Table 6.5: The aggregated comparison matrix of criteria and V values

	C_1			C_2			C_3		
C_1	1.000	1.000	1.000	0.760	1.316	2.590	0.355	0.669	0.760
C_2	0.386	0.760	1.316	1.000	1.000	1.000	0.237	0.508	0.760
C_3	1.316	1.495	2.817	1.316	1.968	4.213	1.000	1.000	1.000

$V(S_{C1} \geq S_{C2}, S_{C3})=0.700$ $V(S_{C2} \geq S_{C1}, S_{C3})=0.446$
$V(S_{C3} \geq S_{C1}, S_{C2})=1.000$

Table 6.6: The aggregated comparison matrix of sub-criteria w.r.t. people criterion and V values

	C_{11}			C_{12}			C_{13}		
C_{11}	1.000	1.000	1.000	1.000	1.316	2.590	0.508	0.760	1.732
C_{12}	0.386	0.760	1.000	1.000	1.000	1.000	0.467	0.669	0.760
C_{13}	0.577	1.316	1.968	1.316	1.495	2.141	1.000	1.000	1.000

$V(S_{C11} \geq S_{C12}, S_{C13})= 0.867$ $V(S_{C12} \geq S_{C11}, S_{C13})=0.521$
$V(S_{C13} \geq S_{C11}, S_{C12})=1.000$

Table 6.7: The aggregated comparison matrix of sub-criteria w.r.t. process criterion and V values

	C_{21}			C_{22}			C_{23}			C_{24}			C_{25}		
C_{21}	1.000	1.000	1.000	1.316	2.236	2.646	0.669	1.316	2.236	0.809	1.316	1.848	0.760	1.495	1.627
C_{22}	0.378	0.447	0.760	1.000	1.000	1.000	0.760	1.316	1.968	0.880	1.136	2.141	0.531	0.669	0.880
C_{23}	0.447	0.760	1.495	0.508	0.760	1.316	1.000	1.000	1.000	0.760	1.732	2.236	0.669	0.760	1.316
C_{24}	0.541	0.760	1.236	0.467	0.880	1.136	1.000	1.000	1.000	1.000	1.000	1.000	0.275	0.386	0.760
C_{25}	0.615	0.669	1.316	1.136	1.495	1.884	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

$V(S_{C21} \geq S_{C22}, S_{C23}, S_{C24}, S_{C25})= 1.000$ $V(S_{C22} \geq S_{C21}, S_{C23}, S_{C24}, S_{C25})=0.630$

$V(S_{C23} \geq S_{C21}, S_{C22}, S_{C24}, S_{C25})=0.711$

$V(S_{C24} \geq S_{C21}, S_{C22}, S_{C23}, S_{C25})=0.465$ $V(S_{C25} \geq S_{C21}, S_{C22}, S_{C23}, S_{C24})=0.674$

Table 6.8: The aggregated comparison matrix of sub-criteria w.r.t. technical criterion and V values

	C_{31}			C_{32}			C_{33}			C_{34}		
C_{31}	1.000	1.000	1.000	1.316	2.590	4.787	1.000	1.968	2.432	1.140	2.141	3.409
C_{32}	0.209	0.386	0.760	1.000	1.000	1.000	0.447	0.577	1.732	0.390	0.760	1.316
C_{33}	0.411	0.508	1.000	0.577	2.236	2.236	1.000	1.000	1.000	0.760	1.316	1.495
C_{34}	0.293	0.467	0.880	0.760	1.316	2.590	0.669	0.760	1.316	1.000	1.000	1.000

$V(S_{C31} \geq S_{C32}, S_{C33}, S_{C34})=1.000$ $V(S_{C32} \geq S_{C31}, S_{C33}, S_{C34})=0.481$ $V(S_{C33} \geq S_{C31}, S_{C32}, S_{C34})=0.697$ $V(S_{C34} \geq S_{C31}, S_{C32}, S_{C33})=0.597$

Finally, by using formula (5.20), we obtain *d'* values for the criteria weights' computation as follows:

$d'(A_{C1}) = min\ V\ (S_{C1} \geq S_{Ci})$= 0.700

$d'(A_{C2}) = min\ V\ (S_{C2} \geq S_{Ci})$= 0.446

$d'(A_{C3}) = min\ V\ (S_{C3} \geq S_{Ci})$= 1.000

For $i \neq 1,2,3$ respectively. Then, the weight vector of criteria is given by the formula (5.21) as

$W'=(d'(A_{C1}), d'(A_{C2}), d'(A_{C3}))$= (0.700, 0.446, 1.000).

Via a normalization, we obtain the normalized weight vectors of the criteria (people, process, technical) with respect to goal as

$W'=(0.326, 0.208, 0.466)^T$.

In a similar fashion, the weight vectors of sub-criteria with respect to criteria can also be calculated. The final results are shown in Table 6.9.

6.2.7 Alternatives' priority weights

In the third stage fuzzy TOPSIS is implemented to determine alternatives' priority weights. The cost, complexity and criticality are considered as the cost criteria while the others are benefit criteria. Same DMs are used for the evaluation process. Again, it is assumed that degrees of the importance for four DMs are equal. Linguistic variables are used for the fuzzy TOPSIS procedure as illustrated in Table 6.10

Table 6.9: Local and global importance of criteria and sub-criteria

Weights				
Dimension	Local Importance	Criteria	Local Importance	Global Importance
C_1	0.326	C_{11}	0.363	0.118
		C_{12}	0.218	0.071
		C_{13}	0.419	0.137
C_2	0.208	C_{21}	0.287	0.060
		C_{22}	0.181	0.038
		C_{23}	0.204	0.042
		C_{24}	0.134	0.028
		C_{25}	0.194	0.040
C_3	0.466	C_{31}	0.360	0.168
		C_{32}	0.173	0.081
		C_{33}	0.251	0.117
		C_{34}	0.215	0.100

.

Table 6.10: Linguistic variables for fuzzy TOPSIS procedure

Linguistic value and triangular fuzzy number	
Linguistic value	Triangular fuzzy number
Very poor	(0, 0, 0.2)
Poor	(0, 0.2, 0.4)
Fair	(0.3, 0.5, 0.7)
Good	(0.6, 0.8, 1.0)
Very good	(0.8, 1.0, 1.0)

The DMs are asked for evaluating the following SLCMs given in Table 6.11 with respect to each sub-criterion.

Table 6.11: Alternatives

Alternative software life cycle models
Waterfall model
V model
Spiral model
Evolutionary Prototyping Model

The evaluation was transformed into fuzzy triangular numbers to evaluate the rating of the life cycle models with respect to each criterion. The alternatives are evaluated in the light of the identified criterion and these linguistic assessments. Then, the evaluations of DMs are aggregated by using arithmetic mean [65]. Results are given in the Table 6.12.

Table 6.12: Aggregated and transformed evaluation values for alternatives

	Comparison of alternatives with respect to sub-criteria											
	WATERFALL			V MODEL			SPIRAL			EVOL. PRO.		
C_{11}	0.450	0.650	0.850	0.375	0.575	0.775	0.450	0.650	0.850	0.300	0.500	0.700
C_{12}	0.225	0.425	0.625	0.450	0.650	0.850	0.600	0.800	1.000	0.700	0.900	1.000
C_{13}	0.150	0.350	0.550	0.375	0.575	0.775	0.450	0.650	0.850	0.375	0.575	0.775
C_{21}	0.450	0.650	0.850	0.300	0.500	0.700	0.225	0.425	0.625	0.075	0.275	0.475
C_{22}	0.375	0.575	0.775	0.150	0.350	0.550	0.075	0.275	0.475	0.075	0.275	0.475
C_{23}	0.000	0.150	0.350	0.375	0.575	0.775	0.375	0.575	0.775	0.650	0.850	1.000
C_{24}	0.075	0.275	0.475	0.525	0.725	0.925	0.600	0.800	1.000	0.700	0.900	1.000
C_{25}	0.650	0.850	1.000	0.375	0.575	0.775	0.450	0.650	0.850	0.225	0.425	0.625
C_{31}	0.150	0.350	0.550	0.750	0.950	1.000	0.375	0.575	0.775	0.700	0.900	1.000
C_{32}	0.375	0.575	0.775	0.375	0.575	0.775	0.375	0.575	0.775	0.450	0.650	0.850
C_{33}	0.300	0.500	0.700	0.375	0.575	0.775	0.450	0.650	0.850	0.525	0.725	0.925
C_{34}	0.225	0.425	0.625	0.600	0.800	1.000	0.375	0.575	0.775	0.450	0.650	0.850

In the second step of fuzzy TOPSIS procudure, the weighted normalized fuzzy decision matrix will be calculated by using equation

$$\tilde{v}_{ij} = \tilde{r}_{ij} \times w_i$$

where w_i are weights of sub-criteria (which are found by fuzzy AHP in the first step) and r_{ij} is the aggregated matrix given in Table 6.12. The weighted normalized fuzzy decision matrix is given in Table 6.13.

In the third step, positive ideal and negative ideal solutions will be identified using the equations (5.26) and (5.27). The fuzzy positive ideal and fuzzy negative ideal solutions are presented in Table 6.14.

Table 6.13: Weighted normalized fuzzy decision matrix

	WATERFALL			V MODEL			SPIRAL			EVOL. PRO.		
C_{11}	0.053	0.077	0.101	0.044	0.068	0.092	0.053	0.077	0.101	0.036	0.059	0.083
C_{12}	0.016	0.030	0.044	0.032	0.046	0.061	0.043	0.057	0.071	0.050	0.064	0.071
C_{13}	0.020	0.048	0.075	0.051	0.079	0.106	0.061	0.089	0.116	0.051	0.079	0.106
C_{21}	0.027	0.039	0.051	0.018	0.030	0.042	0.013	0.025	0.037	0.004	0.016	0.028
C_{22}	0.014	0.020	0.029	0.006	0.013	0.021	0.003	0.010	0.018	0.003	0.010	0.018
C_{23}	0.000	0.006	0.015	0.016	0.024	0.033	0.016	0.024	0.033	0.028	0.036	0.042
C_{24}	0.002	0.008	0.013	0.015	0.020	0.026	0.017	0.022	0.028	0.019	0.025	0.028
C_{25}	0.026	0.034	0.040	0.015	0.023	0.031	0.018	0.026	0.034	0.009	0.017	0.025
C_{31}	0.025	0.059	0.092	0.126	0.159	0.168	0.063	0.097	0.130	0.118	0.151	0.168
C_{32}	0.030	0.046	0.063	0.030	0.046	0.063	0.030	0.046	0.063	0.036	0.053	0.069
C_{33}	0.035	0.059	0.082	0.044	0.067	0.091	0.053	0.076	0.099	0.061	0.085	0.108
C_{34}	0.023	0.043	0.063	0.060	0.080	0.100	0.038	0.058	0.078	0.045	0.065	0.085

Table 6.14: The fuzzy positive ideal and fuzzy negative ideal solutions

Criteria	FPIS(A*)			FNIS(A^-)		
C_{11}	0.053	0.076	0.100	0.036	0.059	0.083
C_{12}	0.050	0.114	0.126	0.016	0.030	0.044
C_{13}	0.020	0.048	0.075	0.061	0.089	0.116
C_{21}	0.004	0.016	0.028	0.027	0.039	0.051
C_{22}	0.003	0.010	0.018	0.014	0.022	0.029
C_{23}	0.028	0.036	0.042	0.000	0.006	0.015
C_{24}	0.019	0.025	0.028	0.002	0.008	0.013
C_{25}	0.026	0.034	0.040	0.009	0.017	0.025
C_{31}	0.126	0.159	0.168	0.025	0.059	0.092
C_{32}	0.036	0.053	0.069	0.030	0.046	0.063
C_{33}	0.061	0.085	0.108	0.035	0.059	0.082
C_{34}	0.060	0.080	0.100	0.023	0.043	0.063

In the next step, the distance of each alternative from A* and A^- and similarities to ideal solution will be calculated by using the vertex equation and equations (5.28) and (5.29) and given in Table 6.15.

Next, the sum of distance of each alternative from positive and negative ideal solutions is computed by using Equation 5.29. The results of these computations are presented inTable 6.16.

Table 6.15: The distance of each alternative from A* and A^-

Waterfall D*	Waterfall D-	V model D*	V model D-	Spiral D*	Spiral D-	Evo. Pro. D*	Evo. Pro. D-
0.0007	0.0178	0.0083	0.0089	0.0007	0.0178	0.0172	0.0000
0.0702	0.0000	0.0553	0.0160	0.0458	0.0267	0.0427	0.0316
0.0000	0.0410	0.0307	0.0102	0.0410	0.0000	0.0307	0.0102
0.0224	0.0000	0.0134	0.0090	0.0090	0.0134	0.0000	0.0224
0.0113	0.0000	0.0028	0.0085	0.0000	0.0113	0.0000	0.0113
0.0283	0.0000	0.0110	0.0174	0.0110	0.0174	0.0000	0.0283
0.0165	0.0000	0.0041	0.0125	0.0023	0.0146	0.0000	0.0165
0.0000	0.0165	0.0104	0.0060	0.0074	0.0091	0.0165	0.0000
0.0931	0.0000	0.0000	0.0931	0.0558	0.0378	0.0069	0.0871
0.0061	0.0000	0.0061	0.0000	0.0061	0.0000	0.0000	0.0061
0.0263	0.0000	0.0176	0.0088	0.0088	0.0176	0.0000	0.0263
0.0376	0.0000	0.0000	0.0376	0.0226	0.0150	0.0150	0.0226

Table 6.16: The total distance of each alternative from A* and A^- and similarity to the ideal solution

	D*	D^-	CC_j*
Waterfall	0.3125	0.0752	0.1940
V Model	0.1598	0.2279	0.5879
Spiral	0.2104	0.1805	0.4618
Evol. Pro.	0.1290	0.2624	0.6704

6.2.8 Determining the final rank and selecting the best SLCM

The last step of the methodology is ranking the models according to their relative closeness to ideal solution. The higher the closeness means the better the rank, so the relative closeness to the ideal solution of the alternatives can be substituted as follows: Evolutionary Prototyping > V Model > Spiral > Waterfall

Evolutionary Prototyping Model has maximum CCj* and that is why it is the best model among the alternative models.

6.3 Results and Sensitivity Analysis

A sensitivity analysis is executed to analyse the two step proposed approach. The idea of sensitivity analysis is to exchange each sub-criterion weight with another sub-criterion weight. In the first step of the sensitivity analysis, sub-criterion in each criterion is exchanged within its criterion. For example, we exchange the weights of ease of management with user involvement and feedback in which both are under the people criterion. Hence, as we have three sub-criteria in people criteria, five in process and four in technical, we need to make [(3)*(3-1)]/2 + [(5)*(5-1)]/2 + [(4)*(4-1)]/2 computations. In brief, 19 computations is done in the first step.

In order to understand the computations easily, each sub-criterion is denoted by numbers given in Table 6.17.

Table 6.17: Numbers for sub-criteria

Number	Sub-Criterion
1	Ease of management
2	User involvement and feedback
3	Cost
4	Complexity
5	Criticality
6	Flexibility
7	Reusability
8	Documentation and software quality
9	Testing and integration
10	Focus on design and architecture
11	Requirements management
12	Formal reviews

The aim of sensitivity analysis is to find CC* values for each computation. CC*1-2 means, the weight of the first sub-criterion is exchanged with the weight of the second sub-criterion. The results of the computations are given in Figure 6.5.

Figure 6.5 summarizes new CC* values of the alternatives. As can be seen in Figure 6.5, Evolutionary Prototyping is still the best alternative in all computations. Although there are some deviations on the values of new CC*, the ranking of the alternatives does not change.

In the second step of the sensitivity analysis, sub-criterion in each criterion is exchanged with another sub-criterion that belongs to different criterion. As there exists 12 sub-criteria, [(12)*(12-1)] / 2 = 66 computations have to be made. However, 19 computations are made in the first step, and 47 computations is done in this step. The results of the computations are given in Figure 6.6.

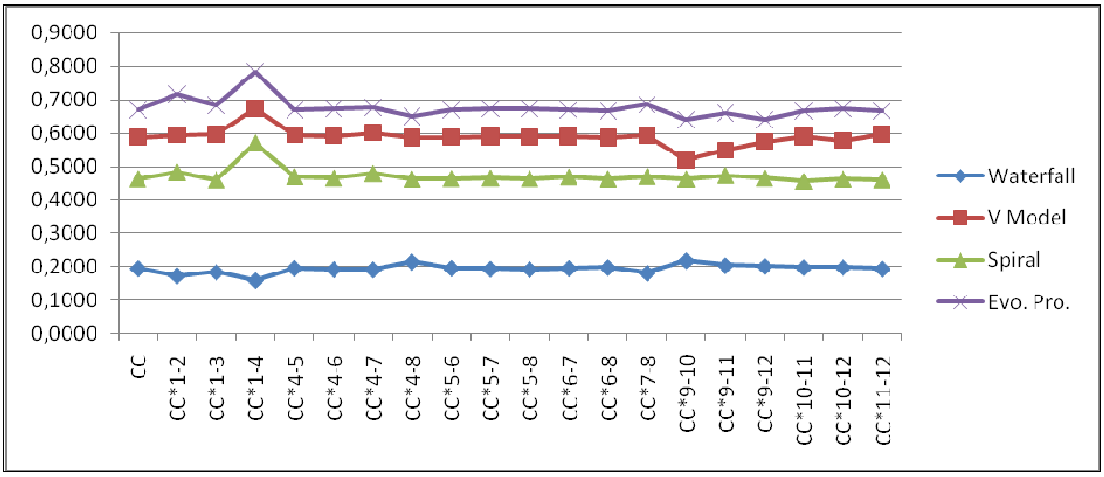

Figure 6.5: Results from first step of sensitivity analysis

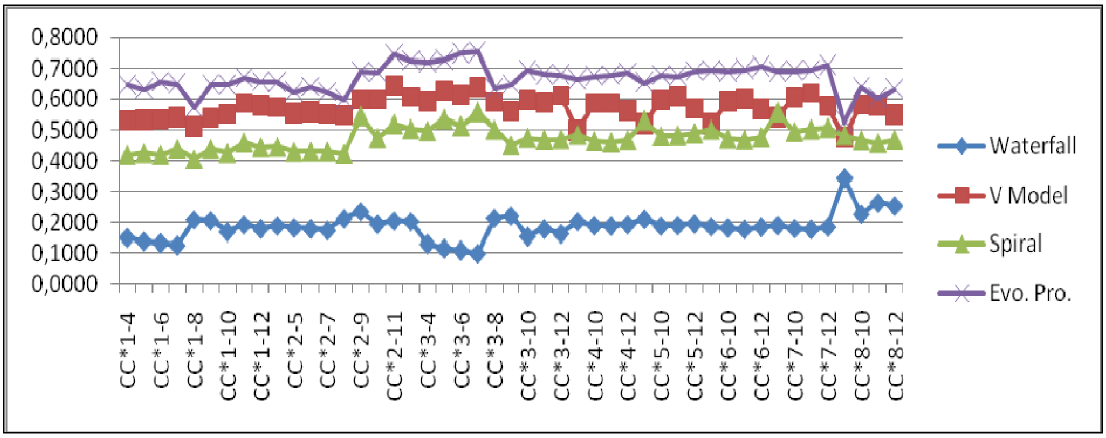

Figure 6.6: Results from second step of sensitivity analysis

Figure 6.6 summarizes new CC* values of the alternatives that are obtained in the second step of the sensitivity analysis. As can be seen from Figure 6.6, Evolutionary Prototyping is still the best alternative in all computations. However, there are some deviations on the values of new CC* and the ranking of the V Model and Spiral is changed in three conditions, where the weight of the 5 is exchanged with 9, 7 with 9 and 8 with 9. Spiral Model is the second in the ranking in these three conditions, while third in the ranking in solution and all other sensitivity analysis computations. Waterfall model is the last in the ranking in all computations as in the solution.

In the study, the weights of the DMs are assumed equal to each other. The weights of the DMs are changed in order to see the deviation and the ranking of the alternatives. The first DM is project manager, and the weight of DM is increased two to five times while others are constant. Besides, the weights of the DM2 and DM3 are increased with DM 1 in several conditions. For example, 2DM1-2DM2-2DM3 indicates that, the weights of DM1, DM2 and DM3 is increased to two times while the weight of the DM4 is constant.

The weights of the criterion that obtained is given in Figure 6.7.

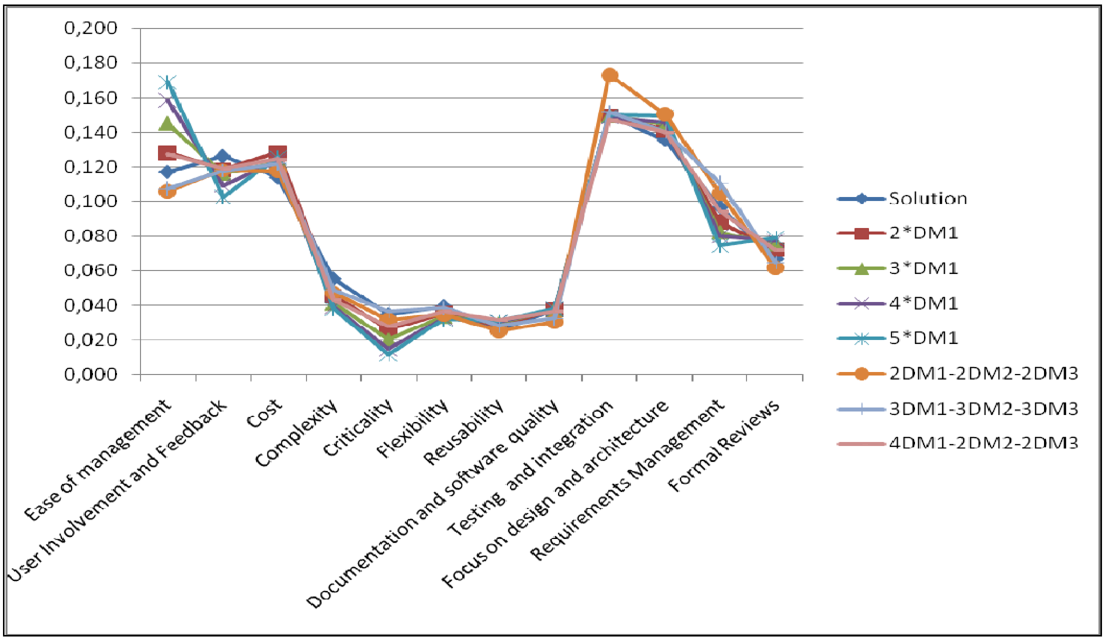

Figure 6.7: Weights of subcriteria obtained by considering different weights for decision makers

As Figure 6.7 shows, the weights of the criterion are sensible to the weights of the DMs. The new weights of 12 criterion are used in 7 different conditions in order to see the best alternative result due to the changes. Again, the Evolutionary Prototyping is the best solution as can be seen in Figure 6.8. There are deviations that cause to change the second rank between V Model and Spiral like in the previous sensitivity analysis computations.

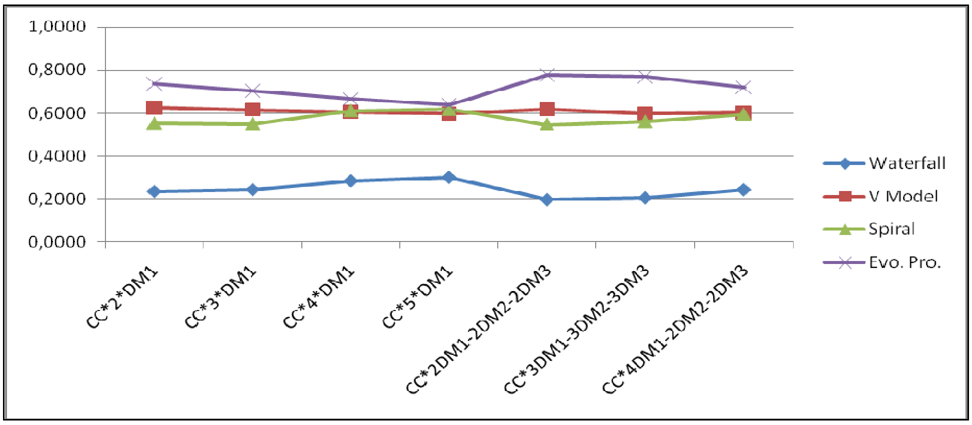

Figure 6.8: Ranking of alternatives with respect to different decision maker weights

7. CONCLUSION AND RECOMMENDATIONS

In this thesis, a fuzzy MCDM approach to SLCM selection problem is proposed. There exists no systematic approach or study about the selection of SLCM by using MCDM methods. The approach proposed tries to fulfill this gap. Organizations may have really serious advantages in competitive software engineering world with the selection of appropriate SLCM, since this selection improves the development process, provides effective utilization of resources and increases productivity and work performance.

Fuzzy AHP and fuzzy TOPSIS are integrated for selection of appropriate SLCM. Fuzzy sets theory enable us to corporate with qualitative and quantitative criteria. Moreover, AHP enables us to deal with large set of criteria as it allows hierarchical structure. Use of AHP also ensures strong and reliable results due to its pairwise comparison feature. TOPSIS lead to make evaluations and calculations simpler and in a rational way.

In this thesis, a wide view of important factors which are considered in SLCM selection is presented. As a result of the study, it is shown that the proposed method is practical for ranking SLCMs with respect to multiple conflicting criteria.

The proposed approach is applied to a specific SLCM selection problem. The application indicated that the approach is useful and can be implemented easily. Moreover, it can be used in a wide variety of application area in software engineering decision making problems using the specific criteria and alternatives set.

As a future direction, other MCDM methods can be included in the SLCM selection problem and the results of different methods can be compared. Moreover, a user-friendly application interface can be developed to speed up and simplify the calculations.

REFERENCES

[1] **Url-1<** http://www1.standishgroup.com/newsroom/chaos_2009.php> accessed at 05.01.2011.

[2] **El Emam K., Koru A. G.,** 2008: A replicated survey of IT software project. IEEE Software. Vol. **25** (5), pp. 84-90.

[3] **Verner, J., Sampson, J., Cerpa, N.,** 2008: What factors lead to software project failure? Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on Digital Object, 10.1109/RCIS.2008.4632095 , pp. 71 – 80.

[4] **Box, G. E. P., and Jenkins, J. M.,** 2008: A guide to the project management body of knowledge (PMBOK GUIDE). Project Management Institute, Inc., Pennsylvania, PA.

[5] **lEEE Std 610.121990**, 1990: IEEE Standard Glossary of Software Engineering Terminology, Standards Coordinating Committee of the Computer Society of the IEEE, New York.

[6] **IEEE Std 12207-2008,** 2008: Systems and software engineering —Software life cycle processes, Software & Systems Engineering Standards Committee of the IEEE Computer Society, New Jersey.

[7] **IEEE Computer Society Professional Practices Committee**, 2004: Guide to the Software Engineering Body of Knowledge SWEBOK, IEEE Computer Society, Los Alamitos, California.

[8] **Ahmad N., Laplante P. A.**, 2006: Software Project Management Tools: Making a Practical Decision Using AHP, IEEE Computer Society Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop SEW-30, 0-7695-2624-1/06.

[9] **Ahmad N., Laplante P. A.**, 2009: Using the Analytical Hierarchy Process in Selecting Commercial Real-Time Operating Systems, International Journal of Information Technology & Decision Making. Vol**. 8** (1), pp. 151-168.

[10] **Tamura Y., Yamada S.**, 2006: Comparison of Software Reliability Assessment Methods for Open Source Software and Reliability Assessment Tool, Journal of Computer Science. Vol. **2** (6) pp. 489-495.

[11] **Peng Y., Kou G., Wang G., Wu W., Shi Y.**, 2011: Ensemble of software defect predictors: an ahp-based evaluation method, International Journal of Information Technology & Decision Making, Vol. **10** (1), pp. 187-206.

[12] **Trienekens J. J. M., Kusters R. J., Brussel D. C.**, 2010: Quality specification and metrication, results from a case-study in a mission-critical software domain, Software Qual J Vol. **18** pp. 469–490.

[13] **Syamsuddin I., Junseok H.**, 2010: The use of ahp in security policy decision making: an open office calc. Application, Journal of Software, Vol. **5 (10)**, pp. 1162-1169.

[14] **Rao R. V., Rajesh T. S.**, 2009: Software selection in manufacturing ındustries using a fuzzy multiple criteria decision making method, promethee, Intelligent Information Management, Vol. **1**, pp. 159-165.

[15] **Kanungo S., Monga I. S.**, 2005: Prioritizing process change requests (pcrs) in software process Improvement, Software Process Improvement and Practıce, Vol. **10**, pp. 441–453.

[16] **Buyukozkan G., Kahraman C., Ruan D.**, 2004: A fuzzy multi-criteria decision approach for software development strategy selection, International Journal of General Systems, Vol. **33 (2–3)**, pp. 259–280.

[17] **Balli S., Korukoglu S.**, 2009: Operating System Selectıon Using Fuzzy Ahp And Topsıs Methods, Mathematical and Computational Applications, Vol. **14 (2)**, pp. 119-130.

[18] **Buyukozkan G., Ruan D.**, 2008: Evaluation of software development projects using a fuzzy multi-criteria decision approach, Mathematics and Computers in Simulation, Vol. **77**, pp. 464–475.

[19] **Thomaidis N. S., Nikitakos N., Dounias G. D.**, 2006: The evaluation of information technology projects: a fuzzy multicriteria decision-making approach, International Journal of Information Technology & Decision Making , Vol. **5 (1)**, pp. 89–122.

[20] **Peng Y., Wang G., Wang H.**, 2010: User preferences based software defect detection algorithms selection using MCDM, Information Sciences, (In press).

[21] **Shyur H. J.**, 2006: COTS evaluation using modified TOPSIS and ANP, Applied Mathematics and Computation, Vol. **177**, pp. 251–259.

[22] **Asosheh A., Nalchigar S., Jamporazmey M.,** 2010: Information technology project evaluation: An integrated data envelopment analysis and balanced scorecard approach, Expert Systems with Applications, Vol. **37**, pp. 5931–5938.

[23] **IEEE Std 1074™-2006**, 2006: IEEE Standard for Developing a Software Project Life Cycle Process, IEEE Computer Society, New York.

[24] **Sommerville I.**, 2004: Software Engineering, Addison-Wesley Publishing Co., 7th Edition.

[25] **ISO 9000: 2005**, 2005: Quality management systems — Concepts and vocabulary.

[26] **Kan S. H.**, 2002: Metrics and Models in Software Quality Engineering, Addison Wesley Publishing Co., 2nd Edition.

[27] **Christensen M. J., Thayer R. H.,** 2001: The Project Managers Guide to Software Engineering's Best Practices, IEEE Computer Society, Los Alamitos, CA.

[28] **Munassar N. M. A., Govardhan A.,** 2010: A Comparison of Between Five Processes Management Model, Advances in Computational Sciences and Technology ,Vol. **3** (3), pp. 277–290.

[29] **Kasse T.,** 2004: Practical Sight into CMMI, Artech House, Boston, London.

[30] **Boehm, B. W.,**1988: A Spiral Model of Software Development and Enhancement, IEEE Computer, Vol. **21 (5)**, pp.61–72.

[31] **Center for Technology in Government University Albany / SUNY**, 1998: A Survey of System Development Process Models CTG.MFA – 003, New York.

[32] **Grady B., Robert C., Newkirk J.,** 1998: Object Oriented Analysis and Design with Applications, Addison Wesley Longman, England.

[33] **Bruegge B., Dutoit A. H.,** 2000: Object-Oriented Software Engineering, Prentice Hall, ISBN-10: 0136066836.

[34] **Hanafiah M., Kasirun Z. M.,** 2007: Using rule-based technique in developing the tool for finding suitable software methodology, Malaysian Journal of Computer Science, Vol. **20 (2)**, pp. 209-224.

[35] **Alexander L. C., Davis A. M.,** 1991: Criteria for Selecting Software Process Models, IEEE, pp. 521-528.

[36] **Kettunen P., Laanti M.**, 2005: How to steer an embedded software project: tactics for selecting the software process model, Information and Software Technology, Vol. **47**, pp. 587–608.

[37] **Sharma A., Gupta A.**, 2011: Critical Risk Factors for Different Project Objectives: A Study of Indian Software Industry, The IUP Journal of Systems Management, Vol. **IX (1)**, pp. 35-55.

[38] **Zadeh, L. A.,** 1965: Fuzzy sets, *Information and Control*, Vol. **8**, pp. 338–353.

[39] **Zadeh, L. A.**, 1975: The concept of a linguistic variable and its application to approximate reasoning, parts 1 and 2, *Information Sciences*, Vol. **8**, pp. 199–249, pp. 301–357.

[40 **Zimmermann, H. J.**, 1991: Fuzzy set theory and its applications (2nd ed.), Kluwer Academic Publisher, London.

[41] **Kaufmann, A., Gupta, M. M.**, 1991: Introduction to fuzzy arithmetic:Theory and applications, Van Nostrand Reinhold, New York.

[42] **Kahraman C.**, 2008: Fuzzy Multi-Criteria Decision Making*: Theory and Applications with Recent Developments Series, Springer Optimization and Its Applications,* Vol. **16**, 590 p.

[43] **Ribeiro R. A., Moreira A. M., Broek P., Pimentel A.**, 2011: Hybrid assessment method for software engineering decisions, *Decision Support Systems,* Vol. **51,** pp. 208–219.

[44] **Davis A. M., Bersoff E. H., Comer E. R.,** 1988: A strategy for comparing alternative software development life cycle models, *IEEE Transactions on Software Engineering,* Vol. **14 (10)**, pp. 1453-1461.

[45] **Saaty T. L.,** 1980: The Analytic Hierarchy Process, McGraw Hill International Book Co., New York.

[46] **Laarhoven P. J. M. and Pedrycz W.**,1983: A fuzzy extension of Saaty's priority theory, *Fuzzy Sets and Systems*, Vol. **11,** pp. 229–241.

[47] **Buckley J. J**., 1985: Fuzzy hierarchical analysis, Fuzzy Sets and Systems, Vol. **17,** pp. 233–247.

[48] **Chang D. Y.**, 1996: Applications of the extent analysis method on fuzzy AHP, *European J. Operational Research*, Vol. **95**, pp. 649–655.

[49] **Kahraman C., Cebeci U., Ruan D.**, 2004: Multi-attribute comparison of catering service companies using fuzzy AHP: The case of Turkey, *International Journal of Production Economics*, Vol. **87**, pp. 171–184.

[50] **Ayag Z., Ozdemir R. G.**, 2006: A fuzzy AHP approach to evaluating machine tool alternatives, *Journal of Intelligent Manufacturing*, Vol. **17**, pp. 179–190.

[51] **Cebeci U., Ruan D.**, 2007: Quality consultants A Multi-Attribute Comparison of Turkish Quality Consultants by Fuzzy Ahp, *International Journal of Information Technology & Decision Making* Vol. **6** (1), pp. 191–207.

[52] **Chang, D. Y.,** 1992: Extent analysis and synthetic decision, *Optimization Techniques and Applications*, Vol. **1**, pp. 352–355.

[53] **Kahraman C., Ruan D., Dogan I.**, 2003: Fuzzy group decision making for facility location selection. *Information Sciences*, Vol. **157**, pp. 135–153.

[54] **Yoon K. P., Hwang C.L.**, 1995: Multiple Attribute Decision Making: An Introduction, Thousand Oaks, Sage Publications, CA.

[55] **Chen C. T.,** 2000: Extensions of the TOPSIS for group decisionmaking under fuzzy environment, *Fuzzy Sets and Systems*, Vol. **114**, pp. 1–9.

[56] **Onut S., Soner S., Efendigil T**., 2008: A hybrid fuzzy MCDM approach to machine tool selection, *Journal of Intelligent Manufacturing*, Vol. **19 (4)**, pp. 443–453.

[57] **Dağdeviren M., Yavuz S., Kılınc N.**, 2009: Weapon selection using the AHP and TOPSIS methods under fuzzy environment, *Expert Systems with Applications,* Vol. **36**, pp. 8143–8151.

[58] **Buyukozkan G., Ruan D.,** 2007: Evaluating government websites based on a fuzzy multiple criteria decision-making approach, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. **15 (3)**, pp. 321-343.

[59] **Onut S., Soner S.,** 2008: Transshipment site selection using the AHP and TOPSIS approaches under fuzzy environment, *Waste Management*, Vol. **28,** pp. 1552–1559.

[60] **Rostamzadeh R., Sofian S.**, 2011: Prioritizing effective 7Ms to improve production systems performance using fuzzy AHP and fuzzy TOPSIS (case study), *Expert Systems with Applications*, Vol. **38,** pp. 5166–5177.

[61] **Sun C.-C.**, 2010: A performance evaluation model by integrating fuzzy AHP and fuzzy TOPSIS, *Expert Systems with Applications*, Vol. **37**, pp. 7745–7754.

[62] **Buyukozkan G., Feyzioglu O., Nebol E.,** 2008: Selection of the strategic alliance partner in logistics value chain, *Int. J. Production Economics,* Vol. **113,** pp. 148–158.

[63] **Amiri, M. P.,** 2010: Project selection for oil-fields development by using the AHP and fuzzy TOPSIS methods, *Expert Systems with Applications,* Vol. **37**, pp. 6218–6224.

[64] **Davis, M. A. P.,** 1994: A Multicriteria decision model application for managing group decisions, *The Journal of the Operational Research Society,* Vol. **45 (1)**, pp. 47-58.

[65] **Detyniecki M.,** 2000: Mathematical aggregation operators and their application to video querying, *PhD Thesis*, Laboratoire d'Informatique de Paris.

[66] **Kwong, C.K. and Bai, H.,** 2003: Determining the importance weights for the customer requirements in QFD using a fuzzy AHP with an extent analysis approach, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University.

[67] **Wang, J. J., Yang, D. L.,** 2007: Using a hybrid multi-criteria decision aid methodfor information systems outsourcing, *Computers & Operation Research*, Vol. **34**, pp. 3691–3700.

APPENDICES

APPENDIX A: Questionnaire for the Evaluation of Software Life Cycle Models

APPENDIX B: An Example MATLAB Program for Computing V values from S_i values

APPENDIX C: Consistency Ratio Computations of Comparison Matrices

APPENDIX D: CC Values obtained in Sensitivity Analysis by Sub-criteria Weight Changes

APPENDIX A: Questionnaire for the Evaluation of Software Life Cycle Models

QUESTIONNAIRE FOR THE EVALUATION OF SOFTWARE LIFE CYCLE MODELS

The aim of this questionnaire is to obtain data for evaluation in ***"A Fuzzy MCDM Approach to Software Life Cycle Model Selection"*** to choose best life cycle model. The questionnaire consists of three parts. In the first part, some questions are presented to gain a brief information about the interviewee/decision-makers. In the second part, the pairwise comparison tables of criteria with respect to goal, and the pairwise comparison tables of sub-criteria with respect to each criteria are given to fill in by decision-makers. In the third part, the alternatives will be evaluated according to each sub-criteria by decision makers.
Thanks for your participation…..

First Part:

1. How many years of experince do you have in software projects?
   - ☐ Up to 5 years
   - ☐ 5 to 10 years
   - ☐ More than 10 years

2. What kind of roles have you taken till now?
   - ☐ System Analyst
   - ☐ Software Architect
   - ☐ Developer
   - ☐ Test Engineer
   - ☐ Project Manager/Leader

3. In what type of projects did you take part?
   - ☐ New Development
   - ☐ Re-Development
   - ☐ Enhancement
   - ☐ Maintenance

4. Which of the following techniques were used during the projects you are involved?
   - ☐ Agile Development
   - ☐ Data Modeling
   - ☐ Multi-Functional Teams
   - ☐ Rapid Application Development
   - ☐ Rational Unified Process
   - ☐ Prototyping
   - ☐ OO Analysis and Design

   Other (specify) ………..

5. Which if any, software process or quality standarts was your projects performed under?
   - ☐ CMMI
   - ☐ SPICE
   - ☐ COBIT
   - ☐ TICKIT
   - ☐ ISO 9002

Second Part:
Before you go on your comparison, it will be useful to have a look at the criteria selected and the definitions of them.

People	Ease of Management	The capability of getting people together to accomplish desired goals and objectives through the software project life cycle that comprise planning, organizing, leading and controlling activities. Predictability, visibility, risk management, communication and coordination are included in this criterion
	User Involvement and Feedback	The participation of the users by evaluating, commenting, rejecting, or approving the product during its development in order to develop a product that meets users' needs
	Cost	Cost related with staff , training, tools, and etc. that will occur during the software development life cycle
Process	Complexity	The degree of difficulty to understand, build and verify of the design or implementation of a process
	Criticality	The degree of impact that a requirement, module, error, fault, failure, or other item has on the development process
	Flexibility	The ease of modification of a system or process for use in applications or environments other than those for which it was specifically designed
	Reusability	The degree of usability of a component, module or any part of the system that was developed in previous development stages in further process
	Documentation and software quality	Documentation are plans, product documents, and the quality is the degree of fullfilment of the customer needs and expectations with the developed software product
Technical	Testing and integration	Combining parts, modules or components together in order to enable to work together in a system, and trying to find any non-conformance in the product developed before it reaches to the end user
	Focus on design and architecture	The degree of emphasis or importance of software design and architecture used for software development process
	Requirements management	The management of customers needs and requirements, adaption of the changing needs to the software environment

	Formal reviews	The control of the document, component or anything that is developed in determined stages (for example, requirements review, design review, document review, and etc.)

In this part of the study, you will make pairwise comparisons using linguistic variables. Linguistic variables is used as they provide a simple way to reason with vague, ambiguous, and imprecise input, and decision makers usually find it is more confident to give internal judgements than fixed value judgements.

Example:
In the pairwise comparison, if you think that **"people"** criteria has **very strong importance** over **"process"** criteria with respect to goal, then put a **X** to the **very strong importance on the left side.**

Or, if you think, **"technical"** criteria has **equal importance** over **"people"** criteria with respect to goal, then put a **X** to the **equal importance** on the right side.

With respect to goal: Software life cycle model selection												
Criteria	Importance of one criteria over another											Criteria
	Extremely preferred	Very strong importance	Strong importance	Weak importance	Equal importance	Just equal	Equal importance	Weak importance	Strong importance	Very strong importance	Extremely preferred	
People		X										Process
People					X							Technical

With respect to goal: Software life cycle model selection												
Criteria	Importance of one criteria over another											Criteria
	Extremely preferred	Very strong importance	Strong importance	Weak importance	Equal importance	Just equal	Equal importance	Weak importance	Strong importance	Very strong importance	Extremely preferred	
People												Process
People												Technical
Process												Technical

With respect to criteria: People												
Criteria	Importance of one criteria over another											Criteria
	Extremely preferred	Very strong importance	Strong importance	Weak importance	Equal importance	Just equal	Equal importance	Weak importance	Strong importance	Very strong importance	Extremely preferred	
Ease of Management												Cost
Ease of Management												User Involvement and Feedback
Cost												User Involvement and Feedback

With respect to criteria: Process												
Criteria	Importance of one criteria over another											Criteria
	Extremely preferred	Very strong importance	Strong importance	Weak importance	Equal importance	Just equal	Equal importance	Weak importance	Strong importance	Very strong importance	Extremely preferred	
Complexity												Criticality
Complexity												Flexibility
Complexity												Reusability
Complexity												Documentation and quality
Criticality												Flexibility
Criticality												Reusability
Criticality												Documentation and quality
Flexibility												Reusability
Flexibility												Documentation and quality
Reusability												Documentation and quality

With respect to criteria: Technical												
Criteria	Importance of one criteria over another											Criteria
	Extremely preferred	Very strong	Strong importanc	Weak importanc	Equal importanc	Just equal	Equal importanc	Weak importanc	Strong importanc	Very strong	Extremely preferred	
Testing and integration												Focus on design and architecture
Testing and integration												Requirements management
Testing and integration												Formal reviews
Focus on design and architecture												Requirements management
Focus on design and architecture												Formal reviews
Requirements management												Formal reviews

Third Part:
In this part of the study, you will evaluate each alternative model(Waterfall, V Model, Spiral, Evolutionary Prototyping) respect to each sub-criterion.

For example, for the first evaluation below,
How well is waterfall model respect to management capability when it is compared with other models? is asked.
You will select linguistic variables given like, very good, good, fair, poor or very poor for your evaluation.

	Sub-criteria	ALTERNATIVES	VERY GOOD	GOOD	FAIR	POOR	VERY POOR
PEOPLE	Ease of management	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	User Involvement and Feedback	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Cost	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					

PROCESS	Complexity	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Criticality	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Flexibility	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Reusability	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Documentation and software quality	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					

TECHNICAL	Testing and integration	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Focus on design and architecture	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Requirements management	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					
	Formal reviews	WATERFALL					
		V MODEL					
		SPIRAL					
		EVOLUTIONARY PRO.					

APPENDIX B: An Example MATLAB Program for Computing V Values from S_i Values

```
function vvalues
sc{1}=[0.153,0.339,0.681];
sc{2}=[0.109,0.203,0.412];
sc{3}=[0.220,0.458,0.981];
for i=1:3;
    for j=1:3;
        if sc{i}(2)>sc{j}(2);
        v(i,j)=1;
        else if sc{j}(1)>=sc{i}(3);
                v(i,j)=0;
            else
                v(i,j)=(sc{j}(1)-sc{i}(3))/(sc{i}(2)-sc{i}(3)-
sc{j}(2)+sc{j}(1));
            end;
        end;
    end;
end;
    v
    v=v';
    minimum=min(v,[],1)
    h=minimum;
    m=0;
    for i=1:3;
    m=m+h(i);
    end;

    for i=1:3;
        h(i)=h(i)/m;
    end;
    norm=h;
    norm
```

APPENDIX C: Consistency Ratio Computations of Comparison Matrices

	People			Process			Technical			
People	1,000	1,000	1,000	0,760	1,316	2,590	0,355	0,669	0,760	
Process	0,386	0,760	1,316	1,000	1,000	1,000	0,237	0,508	0,760	
Technical	1,316	1,495	2,817	1,316	1,968	4,213	1,000	1,000	1,000	
	1	1,43569	0,63162		0,3262		0,91881		2,81671	
	0,79025	1	0,50496		0,2078		0,70084		3,37266	
	1,6858	2,23348	1		0,4659		1,47992		3,17649	
									9,36586	3,12195
								CI	0,06098	
								CR	0,10513	

Figure C. 1: CR computations of the comparison matrix w.r.t. goal

	Ease of management			Cost			User			
Ease of management	1,000	1,000	1,000	1,000	1,316	2,590	0,508	0,760	1,732	
Cost	0,386	0,760	1,000	1,000	1,000	1,000	0,467	0,669	0,760	
User	0,577	1,316	1,968	1,316	1,495	2,141	1,000	1,000	1,000	
	1	1,47572	0,87992		0,363		1,05361		2,90172	
	0,73757	1	0,65032		0,218		0,75837		3,47556	
	1,30161	1,57303	1		0,419		1,23465		2,94806	
									9,32534	3,10845
								CI	0,05422	
								CR	0,09349	

Figure C. 2: CR computations for the comparison matrix w.r.t. people criterion

	Complexity			Criticality			Flexibility			Reusability			Documentation and quality		
Complexity	1,000	1,000	1,000	1,316	2,236	2,646	0,669	1,316	2,236	0,809	1,316	1,848	0,760	1,495	1,627
Criticality	0,378	0,447	0,760	1,000	1,000	1,000	0,760	1,316	1,968	0,880	1,136	2,141	0,531	0,669	0,880
Flexibility	0,447	0,760	1,495	0,508	0,760	1,316	1,000	1,000	1,000	0,760	1,732	2,236	0,669	0,760	1,316
Reusability	0,541	0,760	1,236	0,467	0,880	1,136	1,000	1,000	1,000	1,000	1,000	1,000	0,275	0,386	0,760
Documentation and qu	0,615	0,669	1,316	1,136	1,495	1,884	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	2,15102	1,36152	1,32026	1,39463		0,287		1,40153		4,87828				
	0,48778	1	1,33202	1,26095	0,68097		0,181		0,8938		4,93541				
	0,83032	0,81059	1	1,65402	0,83736		0,204		0,97284		4,75948				
	0,80273	0,85397	1	1	0,42986		0,134		0,8065		6,03666				
	0,76764	1,50028	1	1	1		0,194		1,02384		5,28844				
											25,8983	5,17965			
										CI	0,04491				
										CR	0,04046				

Figure C. 3: CR computations for the comparison matrix w.r.t. process criterion

	Testing and integration			Focus on design and architecture			Requirements management			Formal reviews		
Testing and integratior	1,000	1,000	1,000	1,316	2,590	4,787	1,000	1,968	2,432	1,136	2,141	3,409
Focus on design and ar	0,209	0,386	0,760	1,000	1,000	1,000	0,447	0,577	1,732	0,386	0,760	1,316
Requirements managen	0,411	0,508	1,000	0,577	2,236	2,236	1,000	1,000	1,000	0,760	1,316	1,495
Formal reviews	0,293	0,467	0,880	0,760	1,316	2,590	0,669	0,760	1,316	1,000	1,000	1,000
	1	2,74382	1,88404	2,18461		0,360		1,77948		4,93888		
	0,41886	1	0,74811	0,790252		0,173		0,6823		3,93484		
	0,57394	1,95962	1	1,253247		0,251		1,06749		4,24955		
	0,50701	1,43569	0,83736	1		0,215		0,85717		3,98312		
										17,1064	4,2766	
										CI	0,0922	
										CR	0,10359	

Figure C. 4: CR computations for the comparison matrix w.r.t. technical criterion

APPENDIX D: CC Values obtained in Sensitivity Analysis by Sub-criteria Weight Changes

		Waterfall	V Model	Spiral	Evo. Pro.
solution	CC	0,1940	0,5879	0,4618	0,6704
1-2	CC*1-2	0,1722	0,5949	0,4814	0,7154
1-3	CC*1-3	0,1830	0,5961	0,4588	0,6839
2-3	CC*1-4	0,1576	0,6743	0,5692	0,7826
4-5	CC*4-5	0,1948	0,5947	0,4680	0,6690
4-6	CC*4-6	0,1915	0,5917	0,4639	0,6746
4-7	CC*4-7	0,1906	0,6015	0,4773	0,6762
4-8	CC*4-8	0,2142	0,5869	0,4610	0,6507
5-6	CC*5-6	0,1949	0,5883	0,4626	0,6689
5-7	CC*5-7	0,1926	0,5891	0,4639	0,6728
5-8	CC*5-8	0,1914	0,5888	0,4626	0,6729
6-7	CC*6-7	0,1945	0,5910	0,4673	0,6695
6-8	CC*6-8	0,1967	0,5873	0,4614	0,6676
7-8	CC*7-8	0,1798	0,5939	0,4684	0,6853
9-10	CC*9-10	0,2174	0,5193	0,4613	0,6408
9-11	CC*9-11	0,2028	0,5487	0,4725	0,6610
9-12	CC*9-12	0,2003	0,5746	0,4635	0,6401
10-11	CC*10-11	0,1968	0,5892	0,4542	0,6658
10-12	CC*10-12	0,1970	0,5778	0,4612	0,6730
11-12	CC*11-12	0,1927	0,5970	0,4588	0,6661

Figure D. 1: CC values obtained by sub-criteria weight changes

		Solution	2*DM1	3*DM1	4*DM1	5*DM1	2DM1-2DM2-2DM3	3DM1-3DM2-3DM3	4DM1-2DM2-2DM3
1	Ease of management	0,117	0,128	0,145	0,159	0,169	0,106	0,107	0,127
2	User Involvement and Feedback	0,126	0,118	0,116	0,109	0,103	0,118	0,118	0,119
3	Cost	0,114	0,128	0,123	0,124	0,125	0,118	0,122	0,124
4	Complexity	0,055	0,046	0,041	0,039	0,038	0,047	0,049	0,043
5	Criticality	0,035	0,026	0,020	0,015	0,012	0,031	0,037	0,028
6	Flexibility	0,039	0,035	0,034	0,033	0,032	0,035	0,038	0,037
7	Reusability	0,026	0,029	0,030	0,031	0,031	0,025	0,028	0,032
8	Documentation and software quality	0,037	0,038	0,038	0,038	0,038	0,031	0,033	0,037
9	Testing and integration	0,150	0,150	0,150	0,149	0,150	0,173	0,152	0,147
10	Focus on design and architecture	0,136	0,142	0,145	0,146	0,149	0,150	0,140	0,140
11	Requirements Management	0,097	0,088	0,082	0,080	0,075	0,105	0,111	0,094
12	Formal Reviews	0,067	0,072	0,076	0,078	0,079	0,062	0,065	0,072

Figure D. 2: CC Values obtained DM Weights Changes

		Waterfall	V Model	Spiral	Evo. Pro.
solution	CC	0,1940	0,5879	0,4618	0,6704
1-4	CC*1-4	0,1524	0,5325	0,4213	0,6474
1-5	CC*1-5	0,1407	0,5345	0,4267	0,6337
1-6	CC*1-6	0,1351	0,5352	0,4208	0,6572
1-7	CC*1-7	0,1277	0,5413	0,4407	0,6497
1-8	CC*1-8	0,2099	0,5132	0,4069	0,5719
1-9	CC*1-9	0,2075	0,5402	0,4406	0,6488
1-10	CC*1-10	0,1711	0,5530	0,4267	0,6467
1-11	CC*1-11	0,1937	0,5867	0,4624	0,6699
1-12	CC*1-12	0,1805	0,5805	0,4460	0,6574
2-4	CC*2-4	0,1899	0,5733	0,4478	0,6548
2-5	CC*2-5	0,1834	0,5559	0,4311	0,6242
2-6	CC*2-6	0,1803	0,5588	0,4318	0,6399
2-7	CC*2-7	0,1750	0,5532	0,4322	0,6230
2-8	CC*2-8	0,2135	0,5481	0,4242	0,6001
2-9	CC*2-9	0,2369	0,6001	0,5478	0,6900
2-10	CC*2-10	0,1959	0,5997	0,4760	0,6864
2-11	CC*2-11	0,2061	0,6452	0,5209	0,7448
2-12	CC*2-12	0,2043	0,6085	0,5053	0,7248
3-4	CC*3-4	0,1325	0,5940	0,4987	0,7188
3-5	CC*3-5	0,1174	0,6262	0,5378	0,7273
3-6	CC*3-6	0,1112	0,6141	0,5146	0,7469
3-7	CC*3-7	0,1014	0,6395	0,5601	0,7532
3-8	CC*3-8	0,2153	0,5905	0,5036	0,6349
3-9	CC*3-9	0,2227	0,5608	0,4531	0,6484
3-10	CC*3-10	0,1558	0,5964	0,4771	0,6927
3-11	CC*3-11	0,1795	0,5901	0,4711	0,6805
3-12	CC*3-12	0,1647	0,6117	0,4725	0,6796
4-9	CC*4-9	0,2042	0,5001	0,4864	0,6646
4-10	CC*4-10	0,1909	0,5865	0,4664	0,6757
4-11	CC*4-11	0,1898	0,5860	0,4626	0,6775
4-12	CC*4-12	0,1940	0,5644	0,4696	0,6860
5-9	CC*5-9	0,2121	0,5218	0,5325	0,6541
5-10	CC*5-10	0,1893	0,5980	0,4829	0,6784
5-11	CC*5-11	0,1911	0,6093	0,4849	0,6754
5-12	CC*5-12	0,1964	0,5706	0,4917	0,6907
6-9	CC*6-9	0,1872	0,5216	0,5029	0,6940
6-10	CC*6-10	0,1833	0,5935	0,4743	0,6885
6-11	CC*6-11	0,1788	0,6009	0,4713	0,6960
6-12	CC*6-12	0,1859	0,5681	0,4793	0,7053
7-9	CC*7-9	0,1913	0,5414	0,5595	0,6888
7-10	CC*7-10	0,1811	0,6060	0,4972	0,6921
7-11	CC*7-11	0,1788	0,6208	0,5035	0,6960
7-12	CC*7-12	0,1864	0,5777	0,5095	0,7100
8-9	CC*8-9	0,3452	0,4775	0,4852	0,5271
8-10	CC*8-10	0,2288	0,5832	0,4689	0,6405
8-11	CC*8-11	0,2654	0,5815	0,4598	0,6044
8-12	CC*8-12	0,2560	0,5501	0,4708	0,6326

Figure D. 3: CC values obtained in second step

CURRICULUM VITAE

Candidate's full name: Mümin Hiçdurmaz

Place and date of birth: Balıkesir / 04.12.1983

Permanent Address: Maltepe / İstanbul

Universities and Colleges attended: ITU Industrial Engineering 2006