Fuzzy Based Evaluation of Software Quality Using Quality Models and Goal Models

Arfan Mansoor
Software Architectures and
Product Line Group
Ilmenau University of Technology
Ilmenau, 98693, Germany

Detlef Streitferdt
Software Architectures and
Product Line Group
Ilmenau University of Technology
Ilmenau, 98693, Germany

Franz-Felix Füßl
Software Architectures and
Product Line Group
Ilmenau University of Technology
Ilmenau, 98693, Germany

***Abstract*—Software quality requirements are essential part for the success of software development. Defined and guaranteed quality in software development requires identifying, refining, and predicting quality properties by appropriate means. Goal models of goal oriented requirements engineering (GORE) and quality models are useful for modelling of functional goals as well as for quality goals. Once the goal models are obtained representing the functional requirements and integrated quality goals, there is need to evaluate each functional requirement arising from functional goals and quality requirement arising from quality goals. The process consist of two main parts. In first part, the goal models are used to evaluate functional goals. The leaf level goals are used to establish the evaluation criteria. Stakeholders are also involved to contribute their opinions about the importance of each goal (functional and/or quality goal). Stakeholder opinions are then converted into quantifiable numbers using triangle fuzzy numbers (TFN). After applying the defuzzification process on TFN, the scores (weights) are obtained for each goal. In second part specific quality goals are identified, refined/tailored based on existing quality models and their evaluation is performed similarly using TFN and by applying defuzzification process. The two step process helps to evaluate each goal based on stakeholder opinions and to evaluate the impact of quality requirements. It also helps to evaluate the relationships among functional goals and quality goals. The process is described and applied on 'cyclecomputer' case study.**

***Keywords*—Decision making; Goal Models; Quality Models; NFR; Fuzzy numbers**

I. Introduction

The distinct purpose of software development is to satisfy various stakeholders needs [1]. There are multiple stakeholders involved in the system development and these stakeholders might have different concerns/opinions about the goals to be achieved by the system. Requirements engineering must provide a way to understand stakeholders needs so that high quality software systems are developed. Although stakeholders needs are placed at the most important place, their classification is regarded as the most difficult task. Each stakeholder might have different requirements and sometimes these requirements are of contradicting nature. Therefore, satisfying these requirements is a challenging task [2]. The goal models of goal oriented requirements engineering(GORE) are used to identify and refine the high level goals. Finding the criteria based on GORE require high level goals to be analysed till leaf goals are achieved, that is, until operational requirements are achieved. These leaf level goals are used as criteria for the established high level goals.

There are multiple criteria in one goal model and each criterion may have different importance for various perspective stakeholders, that is, some criteria are more important than others [3]. Stakeholders opinions and preferences should be involved in the process to find the relative importance of each criterion. Normally, there is uncertainty and vagueness about selected criteria because of contradicting stakeholder interests and to find relative importance of criteria according to different stakeholders, one need to perform multi-criteria analysis (MCA). These kind of problems are known as Multi-criteria problems and in general fuzzy set theory is adequate to deal with these problems [4].

Goal Oriented Requirements Engineering (GORE) considers requirements as goals that stakeholders want to be fulfilled. In GORE, goals are refined through AND/OR refinement [5]. By these refinements and by applying heuristics, the functional goals and quality goals are obtained. Functional goals are achieved by operationalization of them either by the system or by external actor while quality goals capture system qualities. The non-functional requirements framework (NFR) [6] deals with the modelling of quality requirements using GORE concepts.

In the context of this paper, GORE is used for identifying and managing the criteria for higher level goals. The leaf level goals help us in establishing the criteria which are used to accumulate stakeholder opinions. These criteria based on stakeholders needs and preferences help to identify the importance of requirements by using qualitative and quantitative reasoning techniques. After the relative importance of each leaf level functional goal, the quality models are used to identify quality goals related to accepted functional goals. Then the impact of quality goals among each other and among functional goals is determined.

The general procedure consists of the following steps:

1) Establishing leaf level functional goals for higher level goals
2) Involving stakeholders opinions
3) Finding scores of each leaf level functional goal
4) Identify quality goals related to functional goals

5) Establish links (contributions) among functional goals and quality goals
6) Measure the impact of quality goals and functional goals
7) Ranking quality goals

GORE is used to explore and establish the leaf level functional goals. These leaf level functional goals are then prioritized based on the stakeholders interests, for determining which of them are more important than others. It serves two purposes:

1) Involving the stakeholders opinions
2) Finding the relative importance

The output of this step is a prioritized list of functional goals. This list is then used to find the impact of quality goals which helps in the evaluation of quality goals among each other and on functional goal.

The remainder of this paper is organized in the following sections: next section gives the literature review on topics used in our approach. Section 3 describes the proposed methodology. Section 4 introduces the 'cyclecomputer' project and gives details of implementing proposed methodology for mentioned project. Section 5 focuses on related work on prioritization and contributions of quality goals on functional goals and vice versa. Finally, last section concludes this paper.

II. Literature Review

A. GORE and Quality Goals

Goal oriented requirements engineering refers to the use of high level goals for requirements elicitation, elaboration, organization, specification, analysis, negotiation, documentation and evolution [5]. One essential output of GORE is goal models. Goal model is a set of goal graphs representing the goals in a top-down or bottom-up hierarchy. Goals are refined into subgoals by using the AND/OR relations. In [5] a catalogue of refinement patterns is proposed. Subgoals describe how the overall goal is achieved. Refinement of a subgoal ends when that subgoal may be associated with a single agent. Most important GORE work includes Non-Functional Requirements framework(NFR) [6], i* framework [7], Goal Oriented Requirements Language (GRL) [8] and Knowledge Acquisitions in automated Specification or Keep All Objects Satisfied (KAOS) [9].

GORE frameworks used the concept of softgoals for quality requirements. Softgoals are goals which can not be fulfilled in their true scene. These are the goals without a clear definition and definite criteria for their fulfilment. Because of their interdependencies and positive/negative influences on each other they are used for handling conflicts and for making trade-offs. Dependencies among the softgoals and their contribution links are useful for the determination of quality goals impact on functional goals [6].

Non-functional requirements are considered from two perspectives [10]:

1) As requirements that describe the properties, characteristics or constraints of the system
2) As requirements that describe the quality attributes the system must have

First type consist of business rules, external interfaces, development constraints and any other requirements that do not describe the functionality of the system. Quality attributes are properties of functional requirements that describe characteristic other than its functionality. An important part of quality attributes is that they should be measurable i.e., one or more metrics can be attached to the quality attribute e.g., response time, throughput time etc. Quality aspects represent the properties of the system that concern stakeholders and these properties affect the degree of satisfaction of the system while constraints unlike qualities are not subject to negotiation, they are off-limits during design trade-offs. [11], [12] argue that quality requirements serve as basis for non-functional requirements in quality models. Quality models used for specifying non-functional requirements provide a hierarchical list of quality attributes also called quality aspects or quality factors. Number of quality models are available in literature. Most of the quality models consist of layers. The number of layers are two (characteristics, sub-characteristics) or three; third layers usually consisting of metrics.

Although these quality model give a systematic structure to quality requirements, they are not consistent with each other [20], for example, understandability is a sub-quality of usability in IS0 9126 [19], but is a sub-class of maintainability in Bohem's model [13]. A comparison of quality models [13], [14], [15], [16], [17], [18], [19] is presented in figure I.

B. Fuzzy Numbers

The functional goals and quality goals help to identify the criteria for the acceptance of target system. There are requirements derived from goal models and quality models which are imprecise in nature. In literature, fuzzy numbers are very popular in engineering disciplines for their ability to represent imprecise and vague information. By using fuzzy sets, requirements are described using linguistic terms. These linguistic terms are then converted into formal representation by using membership functions described for fuzzy numbers [21]. Membership function is the set of real numbers (R) whose range is the span of positive numbers in the closed interval [0,1], where '0' represents the smallest possible value of the membership function, while '1' is the largest possible value [22].

Fuzzy numbers depict the physical world more realistically than single-valued numbers. Among the fuzzy number Triangular Fuzzy Number (TFN) is capable of aggregating the subjective opinions [23]. A triangular fuzzy number (TFN) is described by a triplet (L, M, H), where M is the modal value, L and H are the left (minimum value) and right (maximum value) boundary respectively. TFN is used to represent stakeholder opinions for functional goals and quality goals which are established through goal models and quality models. Fuzziness of TFN is (L, M, H) is defined by the equation 1:

$$TFN(L,M,H) = \frac{H-L}{2M} \tag{1}$$

The membership function $\mu(x)$ for TFN is defined by the

TABLE I: Quality Models Comparison

Factors/Attributes/ Characteristics	Boehm's Model	McCall's Model	Romann Model	Sommerville Model	Dromey's Model	FURPS/ FURPS+	ISO9126 Model
Maintainability	*	*	*		*		*
Flexibility	*	*	*				
Testability	*	*		*			maintainability
Correctness	*	*	*				maintainability
Efficiency	*	*	*	*	*		*
Reliability	*	*	*	*	*	*	*
Integrity	*	*	*		*	*	*
Usability	*	*	*	*		*	
Portability	*	*	*	*	*		*
Reusability	*	*	*		*		
Interoperability		*	*	*		*	
Human Engineering	*		*	*		*	
Understandability	*						
Modifiability	*						maintainability
Functionality			*	*	*	*	*
Performance			*	*		*	
Supportability			*			*	
Clarity	*						
Documentation	*		*	*		*	
Resilience	*						
Validity	*						maintainability
Generality	*		*	*		*	
Economy	*		*				

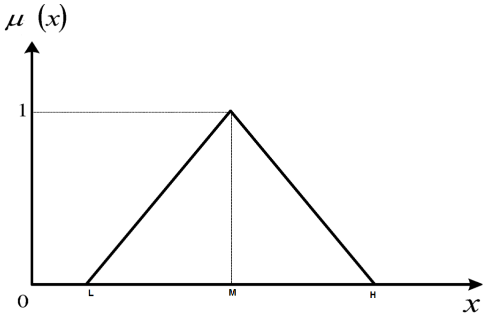

Fig. 1: TFN membership function

equation 2 and is shown in the figure 1 [4] .

$$\mu(x) = \begin{cases} 0, & x < L \\ \frac{x-L}{M-L}, & L \le x \le M \\ \frac{H-x}{H-M}, & M \le x \le H \\ 0, & x > M \end{cases} \tag{2}$$

III. The Proposed Method

In this section, the approach is described on how to use fuzzy number for functional goals and to find out among different functional goals, the ones which lead to better stakeholder satisfaction. After that the impact of quality goals to those functional goals is determined. First of all, higher level goals are modelled and for that, GORE is used to get goal models as a result of it. AND/OR diagrams which are essential output artefact of these goal models are used in the exploration phase of alternatives. The leaf nodes of goal models are used as criteria for functional goals. These criteria are compared based on the weighted scores. The criteria are weighted using fuzzy numbers and stakeholders opinions are taken as input. By using the fuzzy numbers, TFN, we can convert the qualitative information of stakeholders into quantitative one. The proposed methodology consist of following steps and is represented in the figure 2:

1) Establish high level goal(s), refine them using GORE and identify functional goals.
2) Functional goals are refined till we reach on leaf level goals. The leaf level goals are identified as goals directly assignable to agents: either humans or system agents. These are used as establishing criteria for functional goals.
3) Identify relevant stakeholders and take their opinions for above established functional goals as inputs. The input is taken in linguistic terms to ease the stakeholder.
4) Calculate relative importance of each criterion by using fuzzy numbers. TFN are used for aggregating stakeholder opinions.
5) Apply defuzzification process.

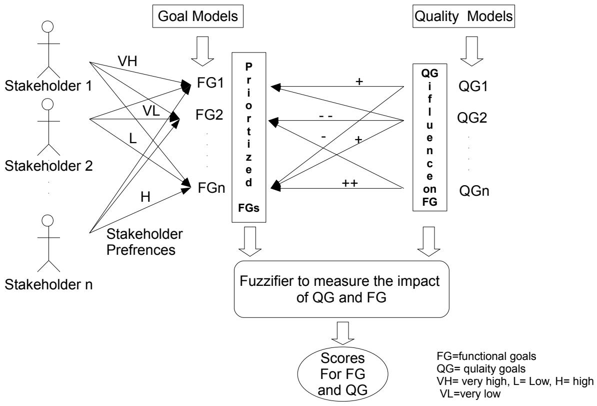

Fig. 2: Proposed methodology

6) Normalize the scores.
7) Identifying quality goals. Use quality models to refine these quality goals.
8) Find the impact of quality goals and functional goals using TFN and defuzzification process.

IV. Case Study

The 'cyclecomputer' system is used as case study for our work which is developed in our research group. The aim of 'cyclecomputer' project is to develop a flexible and modular bicycle computer which is adaptable to the needs of the driver. A driver will be supported while riding the bike, for maintenance issues, for tour preparations, or to enhance the safety using the bike e.g., besides the normal cycling activities one could use the 'cyclecomputer' as a medical device which will support people having of health problems. It can be used for professional cyclist or just for entertainment purposes. A variety of sensors in 'cyclecomputer' provide a comprehensive view of bike, driver/rider and route. In addition to speed, temperature, altitude, geographic location, heart rate; measurements like oil quality and pressure in the damper elements, brake wear or brake fluid quality are relevant to our project. Measurement of the quality framework on strain gauges is also an important requirement. This system will be attached to a bicycle, will process data from various sensors. All data is processed in the 'cyclecomputer' itself or it will communicate with a standard PC in the aftermath of a tour. One of the results of the requirements engineering phase is a goal model [24].

A. Establishing High level Goals

Though there are many goals related to 'cyclecomputer' but for space and simplicity considerations we take following identified high level goal Achieve[TourPlaningServiceSatisfied].

B. Refine Goals to Leaf Levels (establish functional goals)

The above mentioned goal is refined using GORE until they are assignable to agents i.e., human agents or software agents. These leaf levels goals are used as criteria for functional goals. Quality goals which include non-functional requirements and often serve selection criteria are also refined based on quality models. The goals along with their subgoals and short description are presented in table II, while figure 3 shows partial goal model for high level goal Achieve[TourPlaningServiceSatisfied].

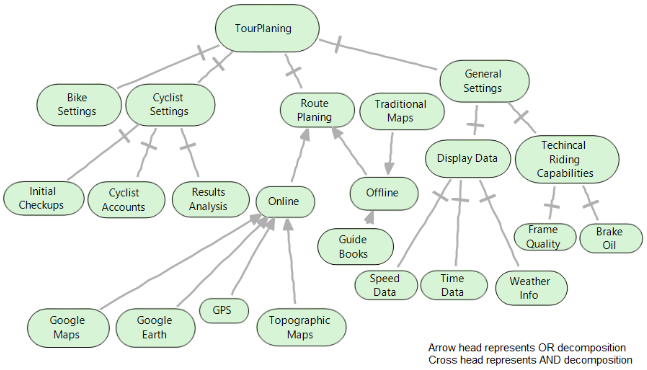

Fig. 3: Partial goal model

C. Stakeholders and Their Opinions

1) Identifying Stakeholders: Though there are number of stakeholders in 'cyclecomputer' but following are the relevant stakeholders for goal Achieve[TourPlaningServiceSatisfied]:

1) Medical Cyclist: People who need a defined training / exercise due to any disease e.g., a heart disease. Medical cyclist can use pulse measurement, blood pressure, calory consumption by 'cyclecomputer' device.
2) Doctor (medical): The doctor will cooperate with a patient to set-up the correct tour plans.
3) Touring Cyclist: People who like to ride the bicycle for long trips (>100km) and they need specific services for their tours. The trips might take more than one day.
4) Analyst: analyse the touring details, analyse the cyclist.

2) Stakeholders Opinions Accumulation: For case study three stakeholders are selected and these stakeholders are asked to give their judgements against functional goals described in table II. Their judgements are used to elicit the importance degree of each functional goal. To enhance the user-friendliness for interacting with stakeholders linguistic terms are used. Linguistic terms are used to describe complex and ill-defined situations which are difficult to be described in quantitative measure. These linguistic terms are represented using TFN. The TFN values for these linguistics terms are derived from [4]. Table III shows the linguistic terms and their representative TFN values. Table IV shows stakeholders judgements against functional goals in table II.

D. Aggregating the Importance Using TFN

The different importance degrees of each functional goal assigned by stakeholders is calculated using TFN. TFN is used to aggregate the subjective opinions of stakeholder using fuzzy set theory. Many methods based on mean, median, min, max,

TABLE II: Partial Goal subgoal description

High level goal	**Sub-goals till functional goals**	**Description**
TourPlanningServiceSatisfied	Route planning	The cycle computer should offer route planning Routing should consider the current weather forecast
	Initial checkups	The cycle computer should offer an initial check-up to assess the drivers capabilities.
	Technical riding capabilities	Frame quality level should be analyzable and visible i.e., show the condition of the frame, interpret the frame condition by a coloured icon. The quality level should be visualized by the time until the frame might break. The cyclist should see the current speed of the cycle. The cyclist should be informed when the oil in the shocks should be changed.
	Weather info	The cyclist should see the current environmental temperature. The temperature of the last 5 days should be analyzable.
	Transferable to web	Track data should be transferred to a Web-portal to enable online competition / comparison.
	Tour details	The cycle computer should provide complete details of the tours. The cyclist should be informed about the current height (above sea level). A cumulative value should be shown by ascended and descended meters.
	Navigation	The cyclist should be able to navigate to a given location. The location could be a point of interest, e. g., a hotel. The cyclist should be informed about his global position on a map.
	Trip suggestions	The cycle computer should offer trip tips for professional sports cyclists e.g., gear change tips, speed tips based on the (known) route.

TABLE IV: Stakeholder judgements

High level goal	**Sub-goals till functional goals**	**Linguistic terms**		
		SH1	**SH2**	**SH3**
TourPlanningServiceSatisfied	Route planning	VH	H	VH
	Initial check-ups	M	H	H
	Technical riding capabilities	H	M	H
	Weather info	L	H	M
	Transferable to web	L	L	M
	Tour details	H	VH	H
	Navigation	VH	H	VH
	Trip suggestions	M	M	M

TABLE III: Linguistic terms and their TFN values

Linguistic terms	**Representative TFN**
Very High	(0.9, 1.0, 1.0)
High	(0.7, 0.9, 1.0)
Medium	(0.3, 0.5, 0.7)
Low	(0, 0.1, 0.3)
Very Low	(0, 0, 0.1)

etc.; are available to aggregate the opinions. Among them average operation is most commonly used aggregation method [3]. Here, the avergae operator is used as an aggregation methods to accumulate stakeholder opinions. Let's say there are 'k' number of stakeholders who assign linguistic term values according to table III to 'n' number of functional goals. The aggregated weight (importance) of each functional goal is calculated as [3]:

$$r_f = \frac{1}{k}\left\{L_f, M_f, H_f\right\} \tag{3}$$

where 'f' represents functional goals from 1...n

$$L_f = \sum_{j=1}^{k} L_{fj}, \quad M_f = \sum_{j=1}^{k} M_{fj}, \quad H_f = \sum_{j=1}^{k} H_{fj} \tag{4}$$

where 'j' represents number of stakeholders from 1...k

E. Apply Defuzzification Process on TFN

After calculating TFN for each functional goal the defuzzification process is applied. Defzzuification process is used to convert calculated TFN values into quantifiable values. Defuzzificatio process is represented by the equation 5 which is derived from [22]:

$$D^{\alpha}\left(x_i\right) = \alpha f_R\left(x_i\right) + \left(1-\alpha\right) f_L\left(x_i\right) \tag{5}$$

where $x_i = TFN_i$ representing triangular fuzzy number. The developer is also involved in the process by representing his preference. α in the above equation represents the preference value of developer and it's value is in the range [0,1].

When $\alpha = 1$ it shows the optimistic view of developer resulting in the equation 6:

$$D^{1}\left(x_i\right) = f_R\left(x_i\right) \tag{6}$$

When $\alpha = 0$ it shows the pessimistic view of developer resulting in the equation 7:

$$D^{0}\left(x_i\right) = f_L\left(x_i\right) \tag{7}$$

Where $f_L\left(x_i\right)$ represents the left end value of TFN$_i$ i.e., pessimistic value. While $f_R\left(x_i\right)$ represents the right end value of TFN$_i$ i.e., optimistic value. These values are represented by the equations 8, 9 respectively:

$$f_L\left(x_i\right) = L_i + \left(M_i - L_i\right)\beta \tag{8}$$

Equation 8 represents left end boundary value.

$$f_R\left(x_i\right) = H_i + \left(M_i - H_i\right)\beta \tag{9}$$

Equation 9 represents right end boundary value.

β in the above equations represents the risks tolerance for particular functional goal and it's value is in the range [0,1]. To keeps things simple we have chosen value 0.5 of preference and risk tolerance against each calculated TFN.

If the only preference value are considered and risk tolerance value is ignored, defuzzification value can be calculated using the equation 10 or 11:

$$D^{\alpha}\left(x_i\right) = \frac{1}{2}\alpha\left(M_i + H_i\right) + \frac{1}{2}\left(1-\alpha\right)\left(L_i + M_i\right) \tag{10}$$

$$D^{\alpha}\left(x_i\right) = \frac{1}{2}\left[\alpha H_i + M_i + \left(1-\alpha\right)L_i\right] \tag{11}$$

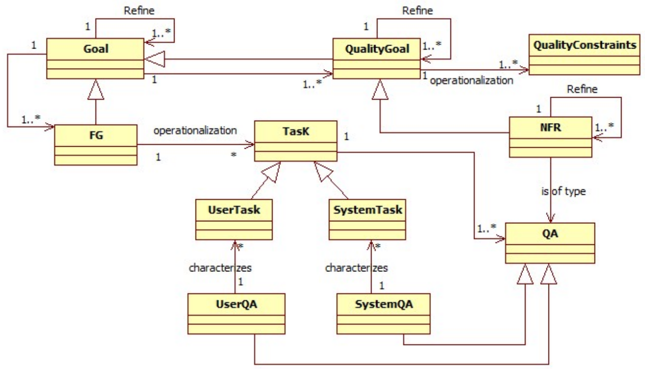

Fig. 4: Conceptual model of functional goal and quality goal integration

F. Normalizing Values Obtained by Defuzzification Process

Although in this paper all the fuzzy numbers are in interval [0,1] and therefore the calculation of normalization is not required, still the scores after the defuzzification process can be normalized by using the equation 12:

$$ND_i = D_i / \sum_{i=1}^{m} D_i \tag{12}$$

where 'm' represents number of functional goals.

Table V represents TFN, defuzzification and final normalized defuzzification values that give the importance of degrees of each functional goal. The defuzzification normalized values give the prioritized list of functional goals.

G. Functional and Quality Goal Impact Measurement

This process consist of three steps:

1) Determining project specific quality goals
2) Determining and evaluating the dependency among quality goals
3) Determining and evaluating the impact of quality goals and functional goals

1) Determining Project Specific Quality Goals: Quality models and NFR framework are useful for determining project based quality goals, that is, the quality goals related to high level system goals. Figure I provides widely used quality attributes in these models. The advantage of using these models is that they provide clear, detail definitions of quality attributes. The universality of these models, because of their acceptance all around the software community. The quality goals are then integrated to functional goal model. Figure 4 represents the conceptual model of quality goals integration to functional goal. This conceptual model is developed by using StarUML.

Figure 5 shows two quality goals 'Safety' and 'Availability' for 'cyclecomputer' functional goal 'RoutePlanning'. These quality goals are represented as softgoals using openOME tool.

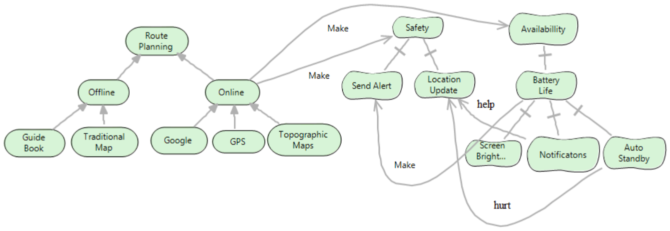

Fig. 5: Quality goals and functional goals

TABLE VI: Linguistic terms and their values for quality goals

Linguistic terms	Numerical Scale
Make	(0.4, 0.5, 0.5)
Help	(0.2, 0.4, 0.5)
Neutral	(-0.2, 0, 0.2)
Hurt	(-0.5, -0.4, -0.2)
Break	(-0.5, -0.5, -0.4)

2) Determining and Evaluating the Dependency between Quality Goals: Quality goals are refined same as functional goals are refined in goal models. These lower level quality goals may influence other quality goals positively or negatively, for example, the fulfilment of one quality goal may hurt or help in the fulfilment of another quality goal. In this step, the importance of each individual quality goal identified in previous step (IV-G1) using TFN (IV-E) is measured and crisp values are obtained by applying the defuzzification process (IV-F). This helps to measure the strength of relationships between quality goals. The linguistic terms and their numerical values used to get crisp values and to measure the relationship strengths are shown in figure VI. The real number interval which represents the direction and strength of relationships among quality goals is set [-0.5,0.5]. The range from negative number is chosen because the contribution 'hurt' or 'break' will have negative impact on other quality goals. These linguistic terms (make, help, hurt, break) are very common in GORE for their use as softgoals contribution. The same linguistic terms are used and numerical values in the range [-0.5,0.5] are defined for these terms.

Let's say there are two quality goals (QG1, QG2) and each goal is refined to four leaf level goals. Now leaf level goals of QG1 are influencing QG2 in positive and/or negative way. Table VII shows their contributions, measurements and final column representing the priority of each leaf level goal of QG1.

The strength of relationships between two quality goals is measured by equation 13. The relationship strength values for goals in table VII are given in table VIII, for example, relationship (LQG1, LQG2, 1.0) gives relationship value (1.0) between leaf level QG1 and leaf level QG2. Here first element LQG1 is impacting or contributing to second element LGQ2 (impacted by LGQ1). These are one-way relationships, the values for (LQG2, LQG1) may be different from (LQG1, LQG2).

$$RS_i = H_i + (M_i - H_i) + L_i + (M_i - L_i) \tag{13}$$

TABLE V: TFN, Defuzzification and Normalized Scores

Functional goals	Linguistic terms	TFN	Defuzzification	Normalized Values		
	SH1	SH2	SH3			
Route planning	VH (0.9, 1.0, 1.0)	H (0.7, 0.9, 1.0)	VH (0.9, 1.0, 1.0)	(0.83, 0.96, 1.0) 0.93	0.17	
Initial checkups	M (0.3, 0.5, 0.7)	H (0.7, 0.9, 1.0)	H (0.7, 0.9, 1.0)	(0.56, 0.76, 0.9)	0.74	0.13
Technical riding capabilities	H (0.7, 0.9, 1.0)	M (0.3, 0.5, 0.7)	H (0.7, 0.9, 1.0)	(0.56, 0.76, 0.9)	0.74	0.13
Weather info	L (0, 0.1, 0.3)	H (0.7, 0.9, 1.0)	M (0.3, 0.5, 0.7)	(0.33, 0.5, 0.66)	0.49	0.08
Transferable to web	L (0, 0.1, 0.3)	L (0, 0.1, 0.3)	M (0.3, 0.5, 0.7)	(0.1, 0.23, 0.43) 0.24	0.04	
Tour details	H (0.7, 0.9, 1.0)	VH (0.9, 1.0, 1.0)	H (0.7, 0.9, 1.0)	(0.76, 0.93, 1.0)	0.9	0.16
Navigation	VH (0.9, 1.0, 1.0)	H (0.7, 0.9, 1.0)	VH (0.9, 1.0, 1.0)	(0.83, 0.96, 1.0)	0.93	0.17
Trip suggestions	M (0.3, 0.5, 0.7)	M (0.3, 0.5, 0.7)	M (0.3, 0.5, 0.7)	(0.3, 0.5, 0.7)	0.5	0.09

TABLE VII: Quality Goals Impact and Measurement

	LQG21	LQG22	LQG23	LQG24	TFN	DFN	Nomalized Score
LQG11	-	Make (0.4, 0.5, 0.5)	Help (0.2, 0.4, 0.5)	Hurt (-0.5, -0.4, -0.2)	(0.03, 0.16, 0.3)	0.15	0.18
LQG12	Make (0.4, 0.5, 0.5)	-	Help (0.2, 0.4, 0.5)	Make (0.4, 0.5, 0.5)	(0.33, 0.46, 0.5)	0.43	0.51
LQG13	Hurt (-0.5, -0.4, -0.2)	Help (0.2, 0.4, 0.5)	-	Help (0.2, 0.4, 0.5)	(-0.03, 0.13, 0.26)	0.11	0.13
LQG14	Make (0.4, 0.5, 0.5)	Help (0.2, 0.4, 0.5)	Hurt (-0.5, -0.4, -0.2)	-	(0.03, 0.16, 0.3)	0.14	0.16

TABLE VIII: Relationship Strength Values

	LQG21	LQG22	LQG23	LQG24
LQG11	-	1	0.8	-0.8
LQG12	1	-	0.8	1
LQG13	-0.8	0.8	-	0.8
LQG14	1	0.8	0.8	-

3) Determining and Evaluating the Impact of Quality goals and Functional goals: In last part of this step, the impact of quality goals and functional goals is determined. Table VI is used to assign the values, impacting goals are arranged vertically and impacted goals are arranged horizontally. Same steps as in IV-G2 are repeated to measure the contributions and relationship strengths.

V. Discussion and Related Work

Success of the software system depends on its capability to satisfy both functional and non-functional requirements. Traditionally, the functional requirements are given high priority while dealing with requirements at abstract level. Goal oriented requirements engineering has been used in representing the requirements at higher level. Goal models combined with quality models can represent both functional and non-functional requirements adequately. However, the impact measurement of contributions among quality goals and also between functional and quality goals is rarely addressed. Because of imprecise nature of the requirements, fuzzy number combined with goal models and quality models can sufficiently represent the requirements impact among each other by quantitative means.

To measure the importance degree of each requirement many requirements prioritizations methods are present in literature. Analytic Hierarchy Process (AHP) is one popular method for prioritization, it involves pair-wise comparison [25]. All pair of requirements are compared to determine the priority level of one requirement over another requirement. Requirements are arranged in matrix form, that is, rows and columns. Then priority is specified to each pair of requirements by assigning a preference value between 1 and 9, where 1 expresses equal value while 9 indicates extreme value. AHP involves stakeholders opinions but pairwise comparison of all requirements make it cumbersome and difficult to use. Proposed methodology in this paper also involves stakeholders opinions and take into consideration both functional and non-functional requirements. Comparisons are made only between the impacting requirements. Importance of both functional and quality goals is obtained using linguistic terms which are easy to deal from stakeholders point of view. These stakeholder opinions are then evaluated using fuzzy set concepts, weight for each functional goals and contribution/impact values are calculated.

In [5] [26] qualitative approaches are used for measuring the contributions. These methods mainly focus on choosing the best alternative. They use temporal logic and label propagation algorithm. We used quantitative approach for measuring the strength of relationships. In [23] prioritizing process for software requirements is highlighted. It considers prioritization of both functional and non-functional requirements at the same level and as a result produces two separate prioritized lists: one of functional requirements and second for non-functional requirements. Like the approach in this paper, their work also used the concepts from [22] but their work is only used for prioritization of functional and non-functional requirement while our work gives an integration model for functional and quality goals and it uses the prioritized requirements to measure their impact on each other.

Wiegers [27] method is semi-quatitative method which focused on customer involvement. Requirements are prioritized based on four criteria defined as benefit, penalty, cost, and risk. The attributes (criteria) are assessed on a scale from 1 (minimum) to 9 (maximum). The customer determines the benefit and penalty values whereas the developers provides the cost and risk values associated with each requirement. Then, by using a formula, the relative importance value of each requirement is calculated by dividing the value of a requirement by the sum of the costs and technical risks associated with its implementation.

The work in [28] focused on modelling the impact of non-functional requirements on functional requirements. For that matter, they investigate the relationships between functional and non-functional requirements. They advocate to define non-functional requirements at the highest level of abstraction like functional requirements. Their proposed approach uses and modifies the NFR framework concepts of contribution but there is nothing mentioned about how to measure the relationships (contributions, impacts) quantitatively.

The work of [21] was the initial attempt to use fuzzy concepts in requirements engineering. Their method deal with conflicting requirements and focus of their work is on prioritizing the conflicting requirements by finding some trade-off between these requirements. The conflicting requirements were represented using fuzzy logic and then they use reasoning scheme to infer the relationship between these conflicting requirements. Ito [2] discussed the uncertainty of design decisions. This work suggests to use AHP and Quality Function Deployment (QFD) for prioritizations and for conflict resolution. In [20] the distinction is made between functional goals and quality goals. They presented non-functional requirements as requirements over qualities i.e., non-functional requirements are modelled as quality goals. For quality goals they use ISO/IEC 25010 standard as reference. They distinguished between domain and co-domain of quality goals. The problem with their model is that functional goal(s) can not be refined into quality goal(s) and vice versa but in GORE we face situations where we encounter these refinements i.e., functional goal refinement results into quality goal and vice versa.

In [29] proposed the guidelines for the elicitation of trustworthy requirements. These guidelines are helpful in selection of project specific quality goals from goal models. Their model consist of three parts: decomposition tree, correlation matrix (CM) and priority vector. Their CM is also base based on fuzzy set theory but it is restricted to elicitation of trustworthy requirements.

In this paper the Fuzzy set concepts are used to evaluate the importance of leaf level functional goal. Weight for each functional goal is calculated based on stakeholders opinions. These weights display stakeholders priorities for all functional requirements. The interaction of stakeholders at early phase of requirements engineering helps to capture the rational (by documenting the preferences) of each requirement and also helps to identify inconsistencies at the early phase of requirements engineering. Using the same method importance weight of quality goals is calculated. Quality goals are tailored using quality models and dependencies among quality goals and functional goals are modelled and measured using fuzzy concepts. The method gives a systematic structure to calculate the fuzzy weight of functional and quality goals. The subjective weights assigned by stakeholders are normalized into a comparable scale. The contributions and strength values are also determined and the strength of the relationships is measured using TFN and defuzzification process.

VI. Conclusion

In this paper an approach is presented to use the goal model of goal-oriented requirements engineering to establish the functional goals as criteria. These leaf levels functional goals are prioritized according to stakeholders preferences. Triangular fuzzy numbers and defuzzification process is used for prioritization, the developers input and risk tolerance is dealt by defuzzification of TFN. After that, the process is used for specified quality goals which are tailored using quality models. In the final step, dependencies among quality goals and between functional goals are evaluated. Therefore, the proposed methodology was used to measure the strength of relationships.

The methodology was explained by 'cyclecomputer' case study where 8 functional goals were established and stakeholders opinions were collected for these functional goals. After calculating the importance value of each functional goal, we integrated quality goals and prioritized them according to their dependencies. This approach is promising for ranking of both functional and quality goals because of stakeholders and developers involvement in the process. The formalization of the approach, goal models and complete quality models integration, implementing it for complete set of non-functional requirements (derived from quality models) and the validation by additional examples are future research topics.

VII. Acknowledgement

We acknowledge support for the Article Processing Charge by the German Research Foundation and the Open Access Publication Fund of the Technische Universit Ilmenau.

References

[1] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *Software, IEEE*, vol. 14, no. 5, pp. 67–74, Sep 1997.

[2] T. Ito, “Dealing with uncertainty in design and decision support applications,” *International Journal of Soft Computing Applications*, vol. Vol.1, no. No.1, pp. pp.5–16,, 2007.

[3] r. Erturul and N. Karakaolu, “Comparison of fuzzy ahp and fuzzy topsis methods for facility location selection,” *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 7-8, pp. 783–795, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00170-007-1249-8

[4] C.-T. Chen, “Extensions of the topsis for group decision-making under fuzzy environment,” *Fuzzy Sets Syst.*, vol. 114, no. 1, pp. 1–9, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1016/S0165-0114(97)00377-1

[5] A. van Lamsweerde, “Requirements engineering in the year 00: a research perspective,” in *ICSE*, 2000, pp. 5–19.

[6] L. Chung, J. Cesar, and S. P. Leite, “Non-functional requirements in software engineering,” 1999.

[7] E. S.-K. Yu, “Modelling strategic relationships for process reengineering,” Ph.D. dissertation, Toronto, Ont., Canada, Canada, 1996, uMI Order No. GAXNN-02887 (Canadian dissertation).

[8] GRL. (2008) Goal-oriented requirement language.

[9] A. Dardenne, A. van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition,” in *Selected Papers of the Sixth International Workshop on Software Specification and Design*, ser. 6IWSSD. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V., 1993, pp. 3–50. [Online]. Available: http://dl.acm.org/citation.cfm?id=171349.156032

[10] A. I. Anton, “Goal-based requirements analysis,” in *Proceedings of the 2Nd International Conference on Requirements Engineering (ICRE ’96)*, ser. ICRE ’96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 136–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850944.853130

[11] X. Franch and J. Carvallo, “Using quality models in software package selection,” *Software, IEEE*, vol. 20, no. 1, pp. 34–41, Jan 2003.

[12] X. Franch, “Systematic formulation of non-functional characteristics of software,” in *Proceedings of the 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice*, ser. ICRE ’98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 174–181. [Online]. Available: http://dl.acm.org/citation.cfm?id=645536.657176

[13] B. W. Boehm, “Software engineering,” *IEEE Trans. Comput.*, vol. 25, no. 12, pp. 1226–1241, Dec. 1976. [Online]. Available: http://dx.doi.org/10.1109/TC.1976.1674590

[14] D. T. Ross and K. E. Schoman, Jr., “Classics in software engineering,” E. N. Yourdon, Ed. Upper Saddle River, NJ, USA: Yourdon Press, 1979, ch. Structured Analysis for Requirements Definition, pp. 363–386. [Online]. Available: http://dl.acm.org/citation.cfm?id=1241515.1241537

[15] G. C. Roman, “A taxonomy of current issues in requirements engineering,” *Computer*, vol. 18, no. 4, pp. 14–23, Apr. 1985. [Online]. Available: http://dx.doi.org/10.1109/MC.1985.1662861

[16] I. Sommerville, *Software Engineering (5th Ed.)*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.

[17] R. G. Dromey, “A model for software product quality,” *IEEE Trans. Softw. Eng.*, vol. 21, no. 2, pp. 146–162, Feb. 1995. [Online]. Available: http://dx.doi.org/10.1109/32.345830

[18] R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[19] I. O. for Standardization, *ISO/IEC TR 9126-4: Software engineering - product quality - Part 4 : Quality in use metrics*. ISO, 2004. [Online]. Available: http://books.google.de/books?id=LM5xkQEACAAJ

[20] F.-L. Li, J. Horkoff, J. Mylopoulos, R. Guizzardi, G. Guizzardi, A. Borgida, and L. Liu, “Non-functional requirements as qualities, with a spice of ontology,” in *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, Aug 2014, pp. 293–302.

[21] J. Yen and W. A. Tiao, “A systematic tradeoff analysis for conflicting imprecise requirements,” in *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, ser. RE ’97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 87–. [Online]. Available: http://dl.acm.org/citation.cfm?id=827255.827815

[22] T.-S. Liou and M.-J. J. Wang, “Ranking fuzzy numbers with integral value,” *Fuzzy Sets Syst.*, vol. 50, no. 3, pp. 247–255, Sep. 1992. [Online]. Available: http://dx.doi.org/10.1016/0165-0114(92)90223-Q

[23] S. P. L. Mohammad Dabbagh, “An approach for integrating the prioritization of functional and nonfunctional requirements,” *The Scientific World Journal*, vol. Volume 2014 (2014), no. Article ID 737626, p. 13 pages, 2014.

[24] A. Mansoor and D. Streitferdt, “On the impact of goals on long-living systems,” in *Software Engineering (Workshops)*, 2011, pp. 133–138.

[25] T. L. Saaty, “Decision making with the analytic hierarchy process,” *Int. J. of Services Sciences,*, vol. Vol.1,, no. No.1, pp. pp.83 – 98, 2008.

[26] J. Castro, M. Kolp, and J. Mylopoulos, “Towards requirements-driven information systems engineering: The tropos project,” *Inf. Syst.*, vol. 27, no. 6, pp. 365–389, Sep. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0306-4379(02)00012-1

[27] K. Wiegers, “First Things First: Prioritizing Requirements,” *Software Development Online*, vol. 7, pp. 48–53, Sep. 1999.

[28] C. Gnaho, F. Semmak, and R. Laleau, “Modeling the impact of non-functional requirements on functional requirements,” in *Advances in Conceptual Modeling*, ser. Lecture Notes in Computer Science, J. Parsons and D. Chiu, Eds. Springer International Publishing, 2014, vol. 8697, pp. 59–67. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-14139-88

[29] F. Shao, R. Peng, D. Sun, H. Lai, and Y.-S. Liu, “An attribute-driven model for trustworthy requirements elicitation,” *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 23, pp. 531–540, 2012.