An Approach to Analyze and Quantify the Functional Requirements in Software System Engineering

M.Karthika,
Assistant Professor
MCA Department
NMSS Vellaichamy Nadar College
Nagamalai,Madurai,
Tami Nadu, India

X.Joshphin Jasaline Anitha
Assistant Professor
MCA Department
NMSS Vellaichamy Nadar College
Nagamalai,Madurai,
Tami Nadu, India

K.Alagarsamy,
Associate professor
Computer Centre
Madurai Kamaraj University
Nagamalai,Madurai,
Tami Nadu, -India

ABSTRACT
Software systems developed now-a- days are by and large more complicated than the existing software. There are certain foundational activities for a system development like the objective of the system, operational requirements, role of hardware and software, the people working in it, database used and the procedures. On understanding the foundational activities based on the System Engineering principle to transform an operational need into more descriptive and will lead to build a right and good product with customer satisfaction. The operational requirements of software system engineering have been classified already .In this paper, the functional requirements classified is common for all the software system developed. Also, it brought out the important role of functional requirements which can effectively be uses to elicit information from the customers more precisely and accurately through Genetic Algorithmic approach. The GA approach identifies, classifies and prioritizes the functional requirements which will provide an insight into the system architecture, also helps to communicate the operational and behavioral characteristics of the new system.

Keywords
System Engineering, Software System Engineering, Functional requirements, product, Evaluation criteria

1. INTRODUCTION

As large systems functions and solutions are dependent on software, System Engineering approach to the development of software will help to avoid the problems associated with the software crisis. These crisis are mainly on the project schedule, cost of estimation, customer satisfaction. A Software product developed may face challenging facts, risks, and issues in managerial aspects such as logistics, lack of co-ordination between the teams developing the product as well as the work process. The need to identify and manipulate the properties of a software system leads to the use of software system engineering. [1]

A system is a group of co-related objects that allows a common goal to be accomplished. In computer systems, these elements include hardware, software, people, facilities, and processes.

System engineering is the practical application of scientific, engineering, and management skills necessary to transform an operational need into a description of a system configuration that best satisfies that need. It is a generic problem-solving process that applies to the overall technical management of a system development project. This process provides the mechanism for identifying and evolving a system's product and process definitions. System engineering involves five functions: Problem definition, Solution analysis, Process planning, Process control and Product evaluation [7].

System engineering provides the baseline for all project development, as well as a mechanism for defining the solution space. The solution space describes the product at the highest level – before the system requirements are partitioned into the hardware and software subsystems.

The developed software has become very larger and more complex than the existing software. The complexity has been increased in variety of phases in the software development life cycle. The understanding of the problem must enforce the below mentioned in system engineering [8].

- Requirement analysis
- Design
- Implementation
- Verification & Validation
- Testing

System Engineering focuses on variety of elements, analyzing, designing and organizing those elements into a system that can become a product, a service or a technology for the transformations or control. Software system engineering manages the technical function of the system products, which produces set of documents. This is a technical process which converts the analytical process into an operational process such as [2],

- Define the problem
- Identify feasible alternatives
- Select the evaluation criteria
- Applying modeling techniques
- Generate input data
- Manipulate the model

This paper categorizes and concentrates only on the functional requirements in the software requirement stage in accordance with the principles of system engineering and its tools.

2. SOFTWARE SYSTEM ENGINEERING

The application of system engineering principle to the development of a computer software system prepares activities, tasks and various set of procedures that is called as

software system engineering (SwSE).This SwSE can be specifically applied in the development of large complex software systems that provides a powerful tool for process and product management. The software system is larger and has high complexity in using it. This will attribute towards the growth in hardware performance as well as the software performance by having reduced software system size which makes the software more complex. The goal of SwSE is to reduce the complexity through several stages of verification and validation.

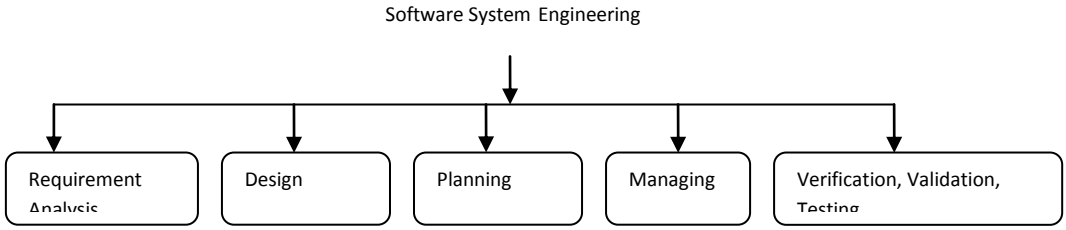


Fig .1. Phases in Software System engineering

Software system Engineering manages the technical function of the system products which produces the documents for reference. Evaluation criteria, a series of steps or an iterative process using genetic algorithms can give this documentation reference.

To achieve a successful process, planning is necessary. Planning tells us the goals in a project the objectives, strategies, polices, plans, and procedure. It tells us in advance what, how, when and who will do the process. To plan a software engineering project we need SWSE management activities that lead in selecting a course of action from alternative possibilities and defining a program for completing those actions.

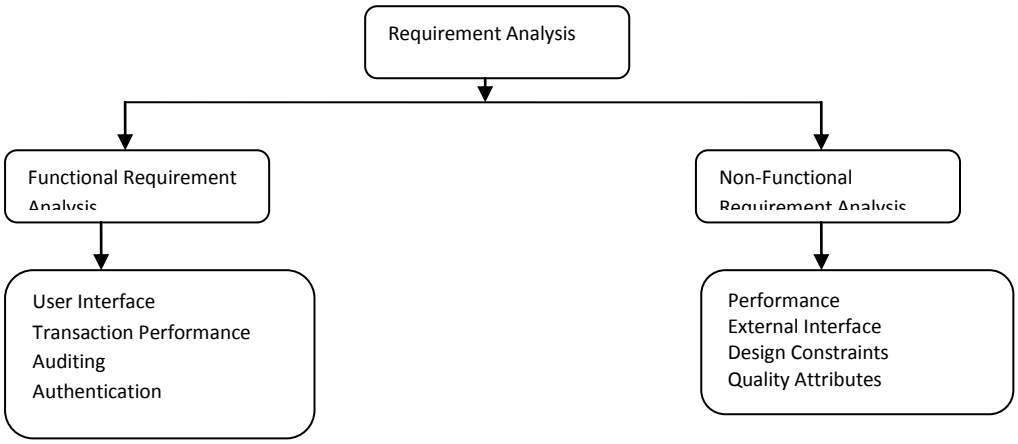
The collection of management activities that used to ensure that the project goes according to the plan is known as process control. The performance and results against plans, notes deviations measured using process control. It takes corrective actions to ensure conformance between plans and actual results. Process control gives us a feedback to the system for how well the project is going. In system, engineering V&V is a continuous process of-monitoring system engineering, SwSE, software engineering, and project management activities to determine that they are following the technical and managerial plans, specifications, standards, and procedures. V&V. evaluate the interim and final product of software engineering.

The user satisfaction must meet out through the system engineers. The success of software development process and the system life cycles undergoes much revaluation process to redefine the functional requirements.

2.1 Requirement Analysis

The first step in any software development activity is to determine and document the system-level requirements in either a system requirements specification (SRS) or a software requirements specification or both. Software requirements include capabilities that a user needs to solve a problem or achieve an objective as well as capabilities that a system or component needs to satisfy a contract, standard, or other formally imposed document. Software requirements analysis begins after system engineering has defined the acquirer and user system requirements. Its functions include identification of all – or as many as possible – software system requirements, and its conclusion marks the established requirements baseline, sometimes called the allocated baseline [3].

Fig.2. Software Requirement Analysis



We can categorize software requirements as shown in Fig.2.Functional requirements specify functions that a system or system component must be capable of performing.

Non-Functional requirements, which include the Performance requirements, specify performance characteristics that a system or system component must possess such as speed, accuracy, and frequency. External interface requirements specify hardware, software, or database elements with which a system or component must interface, or set forth constraints on formats, timing, or other factors caused by such an interface. Design constraints affect or constrain the design of a software system or software system component, for example, language requirements, physical hardware

requirements, software development standards, and software quality assurance standards. Quality attributes specify the degree to which software possesses attributes that affect quality, such as correctness, reliability, maintainability, and portability.

3. FUNCTIONAL REQUIREMENTS

In software system engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements, which impose constraints on the design or implementation, such as performance requirements, security, or reliability. Acceptable systems must satisfy all the mandatory requirements. [12]

Functional requirements are based on the following four classical members have been depicted in Fig .3 as,

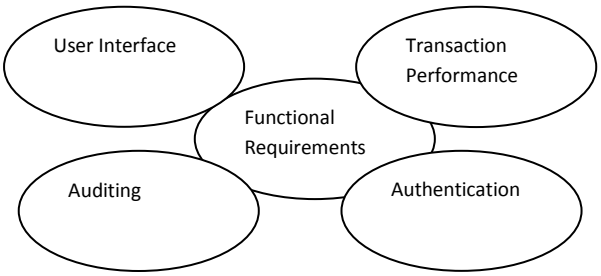


Fig.3. Classification of Functional Requirements.

- Retrieving input from the user
  - Identification
  - Reaction
  - Behavior
- Transaction based performance
  - Identification
  - Reaction
  - Behavior
- Auditing
  - Functions
  - Limitations
- Authentication
  - Members
  - Managers
  - Administrators.

4. MODEL SYSTEM FOR FUNCTIONAL REQUIREMENTS

The representation for the requirement starts from the world view to the design view. This may be a top down or bottom up approach. A system should be modeled, which has an element and also plays a vital role in the development process [9] [10].

Now the modeled developed is top down approach, so it starts form

- Defining the processes
- Represent the behavior of the processes assumptions on which the behavior is based.
- Explicitly define the exogenous end endogenous input to the model.
- Represent all the links.

In constructing a model, the restraining factors such as assumptions, simplifications, limitations and constraints must be considered.

Representing the requirements in World view (WV) as,

$$WV = \{D_1, D_2, D_3 ..., D_n\}$$

$$D_i = D_1, D_2, D_3 ..., D_n$$

D_i are the Set of domains either systems / subsystems.
Representing the requirement as Domain view (DV),

$$DV = \{E_1, E_2, E_3, ...E_m\}$$

$$E_j = E_1, E_2, E_3, ..., E_m$$

E_j are the set of elements which satisfies the objective or the goal of accomplished tasks.

Representing the Elemental view (EV) as,

$$EV = \{C_1, C_2, C_3, ...C_k\}$$

$$C_k = C_1, C_2, C_3, ..., C_k$$

C_k are the set of technical components that achieves the necessary function of an element [3].

These assumptions shown in Fig.4, of world, domain, elementary and components of the can be evaluated via a genetic algorithmic approach.

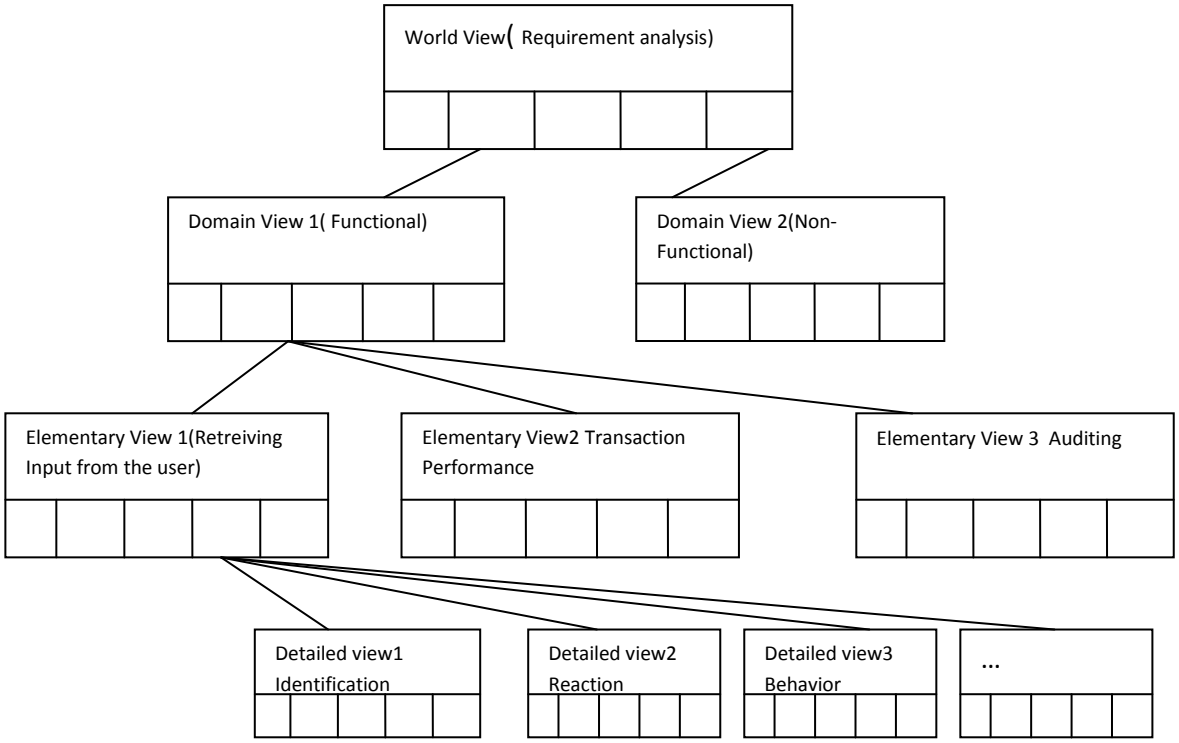


Fig. 4. A Model Representation - Functional requirements

4. EVALUATION OF REQUIREMENTS THROUGH GENETIC ALGORTIHM

4.1 Genetic algorithm

Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution; especially those follow the principles first laid down by Charles Darwin of "survival of the fittest". Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones [4].

Genetic algorithm is a stochastic random global search and optimization method that minimizes and produces variety of solutions, based on evaluation. The three most important aspects of using genetic algorithms are: (1) definition of the objective function, (2) definition and implementation of the genetic representation, and (3) definition and implementation of the genetic operators. On defining these three, the generic genetic algorithm should work fairly well. Beyond that you can try many different variations to improve performance, find multiple optima, or parallelize the algorithms [5].

Also, in searching a large state-space, multi-modal state-space, or n-dimensional surface, a genetic algorithm may offer significant benefits over more typical search of Optimization Techniques. (linear programming, heuristic, depth-first, breath-first, and praxis.)

4.2 An Genetic Algorithmic Approach

GAs is an effective strategy, applied in the functional requirement analysis phase. This allows the retention of existing modeling and simulation tools for building the objective functions and allows the user to make direct comparisons between genetic methods and traditional procedures, which will produce n number of solutions from four different requirements of type functional [6].

The above-mentioned Modeled system developed for functional requirements such as world, domain, elementary and component views were the inputs given to the GA tool. The worldview represents overall software requirements. Domain view representation shows the functional requirement such as user input, transaction performance, Auditing and Authentication. The Elementary view has identification, reaction, behavior, functions, limitations and people of the domain values.

Now let us take the domain view to the GA tool , which will produce a set of combinations through crossover operator, under repeated set of iterations and the selection tool produces the best combination of domain .This selection will help to go through the next design phase.

4.2.1 Implementation Details

The GA approach in this paper will produce n (n+1) combination of requirements for n samples or requirements given and we can choose the necessary combinations of requirement for the product to be developed based on Natural Selection.

After an initial requirement generated randomly, the algorithm evolves the through three operators:

- **Selection** which equates to best of the world view requirements;
- **Crossover** which represents combination between requirements;
- **Mutation**, which introduces random modifications.

This is done using GA tool of MATLAB, which produces various combinations of domain, Elementary and also develops a new set of requirements (the mutants).

5. CONCLUSION

Software System Engineering performs a variety of functions at various stages in the product development life cycle. Software Requirement Analysis is a phase in which reevaluation has to be started. Requirements are classified into functional and non-functional requirements relevant to the product or process developed. The functional requirements categorized as, user interface, transaction performance, auditing and authentication were done in accordance with their functions. The basic representation which is relevant to software system engineering such as world, domain, elementary and component view were discussed. The views noted have been approached from top - down. This paper focuses on using the GA tool to have an iterative process of identifying and refining the requirements by a method of reevaluation. A series of repeated steps using the tool will help us to elicit all the requirements to develop the design phase successfully. The implementation through the tool will be discussed in the future.

6. REFERENCES

[1] A.Terry Bahill, Clark Biggs, " The Systems Engineering started in the middle process: a consensus of system Engineers and project Mangers, 2001, John Wiley & sons, Inc.syst Eng:156 – 157.

[2] Eushiuan Tran , " Requirements & Specification"

[3] Quantifying Software Reliability Readiness Astana, A.; Olivieri, J. ; Alcatel - Lucent, Westford, MA, USA

[4] Software Engineering a Practioner's Approach, sixth Edition, Roger S.Pressman.

[5] Software Engineering Theory and Practice, Fourth Edition, Shari Lawerence Fleeger, Joanne M. Atlee.

[6] A. J. Chipperfield and P. J. Fleming, the MATLAB Genetic Algorithm Toolbox.

[7] Richard H.Thayer, "Software System Engineering: A Tutorial"IEEE 2002, pp 68-73.

[8] Eric C.Honour, " Understanding the Value of System Engineering"

[9] "Introduction to Systems Engineering"

[10] Linda Westfall, "Software Requirements Engineering: What, Why, Who, When, and How", 2005-2006.

[11] Dr.Judith Dahmann, "Importance of systems engineering in Early Acquisition".

[12] www.ofnisystems.com/Validation/Functional_Requirements.htm

[13] Terry Bahill, "The Eight System Engineering Documents", 2001-2009.

[14] A.Terry Bahill, Frank F. Dean, "What is system engineering? A consensus of senior system Engineers", Jan 15, 2009.

[15] Dr. Laurie Williams, " A (Partial) Introduction to Software Engineering Practices and Methods", NCSU CSC326 Course Pack, 2008-2009 (Fifth) Editions.

[16] Prof. Dr. Oscar Nierstrasz, Dr. St´ephane DucasseMichele Lanza, "Recovering the Evolution of Object Oriented SoftwareSystems Using a Flexible Query Engine", June 2001.

[17] Ruth Malan and Dana Bredemeyer "Functional Requirements and Use Cases".

[18] http://media.wiley.com/product_data/excerpt/96/07695119/0769511996.pdf.