CSc 322 - Software Engineering
Lab – Spring 2018

Teaching Assistant: Isatou Sanneh
Email: isanneh00@citymail.cuny.edu

Identifying the Requirements from Problem Statements

February 9th, 2018

Introduction

Requirements identification is the first step of any software development project. Until the requirements of a client have been clearly identified, and verified, no other task (design, coding, testing) should begin. Usually business analysts who have domain knowledge on the subject matter discuss with clients and decide what features are to be implemented.

In this session, we will learn how to identify functional and non-functional requirements from a given problem statement. A problem statement is a clear and concise description of an issue to be solved. Functional and non-functional requirements are the primary components of a Software Requirements Specification.

Requirements: Requirements specify how the target system should behave. It specifies what to do, but not how to do it. Requirements Engineering refers to the process of understanding what a customer expects from the system to be developed, and to document them in a standard and easily readable and understandable format. This documentation will serve as reference for the subsequent design, implementation and verification of the system.

It is necessary and important that before we start planning, designing, and implementing a software system for our client, we are clear about its requirements. If we don't have a clear vision of what is to be developed and which features are expected, there would be serious problems, and customer dissatisfaction as well.

Characteristics of Requirements: Requirements gathered for any new system to be developed should exhibit the following three properties:

- **Unambiguity:** There should not be any ambiguity on what a system to be developed should do. For example, consider you are developing a web application for your client. The client requires that enough number of people should be able to access the application simultaneously. What's the "enough number of people"? That could mean 10 to you, but, perhaps, 100 to the client. There's an ambiguity.
- **Consistency:** To illustrate this, consider the automation of a nuclear plant. Suppose one of the clients say that if the radiation level inside the plant exceeds R1, all reactors should be shut down. However, another person from the client side suggests that the threshold radiation level should be R2. Thus, there is an inconsistency between the two end users regarding what they consider as the threshold level of radiation.

- **Completeness:** A particular requirement for a system should specify what the system should do and also what it should not. For example, consider a software to be developed for an ATM. If a customer enters an amount greater than the maximum permissible withdrawal amount, the ATM should display an error message, and it should not dispense any cash.

Types of Requirements

Based on the target audience or subject matter, requirements can be classified into different types such as:

- User requirements: They are written in natural language so that customers can verify their requirements have been correctly identified.
- System requirements: They include technical terms and/or specifications, and are meant for the development or testing teams.

Functional and Non-functional Requirements

Requirements can be classified into two categories based on what they describe:

- Functional requirements (FRs): These describe the functionality of a system – how a system should react to a particular set of inputs and what should be the corresponding output.
- Non-functional requirements (NFRs): They are not directly related to what functionalities are expected from the system. However, NFRs could typically define how the system should behave under certain situations.

Non-functional requirements could be further classified into different types:

- Product requirements: For example, a specification that the web application should use only plain HTML, and no frames
- Performance requirements: For example, the system should remain available 24x7

Example Problem Statement – Library Information System (LIS) for an Institute

LIS will enable its members to borrow a book (or return it) with ease while sitting at their desk/chamber within the institution. The system also enables a member to extend the date of his/her due date if no other requests have been made for that particular book. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books. However, issuing or returning books is restricted to valid users (members) of LIS only.

The final deliverable would a web application, which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (eg., passwords) is stored in plain text.

Identification of Functional Requirements

- New user registration: Any member of the institute who wishes to use the facilities of the library has to register for an account with the Library Information System. On successful registration, a user ID and password would be provided to the member. He/She has to use this credential for any future transaction in LIS.

- Search book: Any member of LIS can use this facility to check whether any particular book is present in the institute's library. A book could be searched by its: Title, Author's name, Publisher's name
- User login: A registered user of LIS can login to the system by providing his user ID and password as set by him/her during registration. After a successful login, the user will be redirected to the homepage where he/she can access the different functionalities of LIS: search book, issue book, return book, reissue book. Any user ID not registered with LIS cannot access the homepage – a login failure message would be shown to him/her, and the login dialog would appear again. This same thing happens when any registered user provides an incorrect password. However, if an incorrect password has been provided for three times consecutively, the security question for the user (specified during registration) with an input box to answer it are also shown. If the user answers the security question correctly, a new password would be sent to his/her email address. In case the user fails to answer the security question correctly, his/her LIS account would be blocked. He/she will have to contact the administrator in order to unblock the account.
- Borrow/Issue book: Any member of LIS can borrow a book from the library provided that:
    - – The book is available in the library i.e. could be found by searching for it in LIS
    - – No other member has currently borrowed the book
    - – Current user has not borrowed the maximum number of books that a user can checkout at any given time

  If the above conditions are met, the book is issued to the member. Note that this FR would remain incomplete if the "maximum number of books that can be issued to a member" is not defined. We assume that this number has been set to four for students and research scholars, and ten for professors. Once a book has been successfully issued, the user account is updated to reflect the same.
- Return book: A book is issued for a finite time, which we assume to be a period of 20 days. That is, a book borrowed from the library should be returned within 20 days by the corresponding LIS member. After successful return of a book, the user account is updated to reflect the same.
- Reissue/Extend book: Any member who has issued/borrowed a book may decide to extend the book, and get permission to keep it for an additional 20 days. However, a member can only extend any book at most twice, after which it has to be returned. Once a book has been successfully extended, the user account is updated to reflect the information.

Identification of Non-functional Requirements

- Performance Requirements: This system should remain accessible 24x7; At least 50 users should be able to access the system simultaneously at any given time
- Security Requirements: This system should be accessible only within the institute LAN; The database of LIS should not store any passwords in plain text – a hashed value has to be stored

Practice Task

Identify and list the functional and non-functional requirements

Consider the problem statement for an "Online Auction System" to be developed:

> New users can register to the system through an online process. By registering, a user agrees to abide by different predefined terms and conditions as specified by the system. Any registered user can access the different features of the system authorized to him/her after authentication (successful login). An authenticated user can put items in the system for auction. They can also place a bid for an item. Once the auction is over, the item will be sold to the user with the maximum bid. Payments are to be made by third party payment services, which, of course, is guaranteed to be secure. The user selling the item will be responsible for its shipping. If the seller thinks he’s getting a good price, he can, however, sell the item at any point in time to the maximum bidder available.