**Definition & Planning**

**Requirements Analysis**

Requirement Analysis is done in order to understand the problem the software system is to solve.

There are two major activities in this phase:  problem understanding or analysis and requirement specification. In problem analysis, the aim is to understand the problem and its context, and the requirements of the new system that is to be developed. Once the problem is analyzed and the essentials understood, the requirements must be specified in the requirement specification document. The requirements document must specify all functional and performance requirements; the formats of inputs and outputs etc.

**Software  Design :**

The purpose of design phase is to plan a solution of the problem Specified by the requirements document. This phase is the first step in moving from the problem domain to solution domain.

The design activity often results in three separate outputs: architecture design, high level design and detailed design.

**Coding  :**

The main goal of coding phase is to translate the design of the System into code in a given programming language. The coding phase affects both testing and maintenance profoundly. The goal of coding should be to reduce the testing and maintenance effort, hence during coding the focus should be on developing programs that are easy to read and understand.

**Testing  :**

The function of testing is to detect the defects in the Software. The main goal testing is to uncover requirement, design and coding errors in the programs.

The main goal of the requirement phase is to produce the software requirement specification (SRS), which accurately capture the client’s requirements. SRS is A document that describes what the software should do. The basic purpose of SRS is to bridge the communication gap between the clients, the end  users  and the Software developers. Another purpose is helping user to understand their own needs.

**Purpose:**

The BLS typically contains the brief description of the project. The purpose of the requirement document is to specify all the information required to design, develop and test the software.

The purpose of this project is to provide a friendly environment to maintain the details of books and library members.

The main purpose of this project is to maintain easy circulation system using computers and to provide different reports.

**Software Requirements Specifications**

A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).Recommended approaches for the specification of software requirements are described by IEEE 830-1998. This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

**Types of Requirements are categorized in several ways.**

The following are common categorizations of requirements that relate to technical management:

**Functional Requirements**

Functional requirements explain what has to be done by identifying the necessary task,

action or activity that must be accomplished. Functional requirements analysis will be

used as the to plevel functions for functional analysis.

**Non-functional Requirements**

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors.

**Performance Requirements**

The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements.

Design Requirements

The ―build to,‖ ―code to,‖ and ―buy to‖ requirements for products and ―how to execute‖ requirements for processes expressed in technical data packages and technical manuals.

**Derived Requirements**

Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.

**Allocated Requirements**

A requirement that is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items.

**Functional Requirements**

Systems engineering model of Specification and Levels of Development. During system development a series of specifications are generated to describe the system at different levels of detail. These program unique specifications form the core of the configuration baselines. As shown here, in addition to referring to different levels within the system hierarchy, these baselines are defined at different phases of the design process.

**Functional specification topics**

**Purpose**

There are many purposes for functional specifications. One of the primary purposes on team projects is to achieve some form of team consensus on what the program is to achieve before making the more time-consuming effort of writing source code and test cases, followed by a period of debugging. Typically, such consensus is reached after one or more reviews by the stakeholders on the project at hand after having negotiated a cost-effective way to achieve the requirements the software needs to fulfill.

**Process**

In the ordered industrial software engineering life-cycle (waterfall model), functional specification describes what has to be implemented. The next system specification document describes how the functions will be realized using a chosen software environment. In not industrial, prototypical systems development, functional specifications are typically written after or as part of requirements analysis. When the team agrees that functional specification consensus is reached, the functional spec is typically declared "complete" or "signed off". After this, typically the software development and testing team write source code and test cases using the functional specification as the reference. While testing is performed the behavior of the program is compared against the expected behavior as defined in the functional specification

**Types of software development specifications**

 Advanced Microcontroller Bus Architecture

 Bit specification

 Design specification

 Diagnostic design specification

 Multiboot Specification

 Product design specification

 Real-time specification for Java

Software Development Process – activities and steps Sofia 2010 Page 11 of 42

 Software Requirements Specification

**See also**

 Benchmarking

 Benchmark specification

 Extensible Firmware Interface

 Software development process

 Specification (technical standard)

 Verification and Validation (software)

**Task Distribution**

The idea of task distribution seems to be very simple: there can be a group of employees and each employee needs to do assigned tasks, while the group manager needs to distribute tasks according to skills, abilities, workload and duties of employees. In real practice, realization of this idea may become challengeable because the group manager needs to know real skills of employees, clearly understand their abilities, estimate workload and consider existing duties in order to distribute and assign tasks appropriately.

To help you with making task distribution, this review shows that distributing task can be represented as a process that has several phases and that can be managed by using task management software. Such software will help create diagrams, prepare total task presentations, and make critical selections.

**The process of distribution consists of the following phases:**

**selection** – During the first phase of the task distribution process, you think over the work you need to get completed. For example, you can be a project manager responsible for completion of project work. You start selecting jobs and tasks to plan and schedule the project. The task selection phase results in creation of tasks, to-do lists and job lists.

**presentation** – When you have defined the work and divided it into tasks and jobs, your next step is to prepare task presentation. The purpose of the presentation phase is to show and explain employees what they are expected to do, what tasks and jobs are selected, and within what timeframes the work is planned being accomplished. The task presentation phase is very important for team building, and you as a project manager will pass through this phase when running the project keep-off meeting.

**dispatching** – This phase is critical to defining internal/external duties of employees and delegating tasks. After tasks and jobs have been presented to employees involved in your project, you need to dispatch descriptions of the project work to project resources. During the task dispatching phase, you take care of timely sending and delivering of tasks and jobs, and task management software will be helpful as it supports collaboration between users, immediate information exchange, and secure project data sharing.

**authorization –** Once all assigned resources have received descriptions of the project work and tasks have been dispatched, your next step is to authorize tasks. The task authorization phase means local managers and team leaders (often in outsourced projects) make sure that all tasks and jobs are assigned to appropriate resources. After inspection of resource assignments is completed, local managers and team leaders report to senior management, and the end of the task authorization can be formally considered as the start point of project implementation.

**outsourcing** – This is an optional phase of the process, as not every project is fulfilled (partly or completely) by outsourced resources. The outsourcing phase means you consider outsourcing as an opportunity to hire foreign human resources which will do tasks and jobs of your project. Often outsourced projects are effective enough but less predictable and manageable, so the phase should be considered carefully. The phase is closely linked to and dependent on the dispatching phase, because outsourced resources need to receive clear, unambiguous and timely information about assigned tasks and jobs.