In this note we give the requirements process in a software organization, a template for the requirements document, and the process to manage changes to the requirements. This note is based on my book *CMM in Practice* (Addison Wesley, 1999). The book also gives the requirements of an example project executed in this company.

Requirements Analysis and Specification

**Overall Process**

The activities that are performed during the requirements phase largely focus on two areas – problem analysis and product description. In the overall process shown below, the problem analysis activities are grouped into three phases – preparing, gathering requirements, and analysis. The product description activities are also grouped into three phases – preparing the SRS, reviewing it, and obtaining the final sign-off from the customer. After the sign-off, the initial baseline for requirements is established.

Though the phases are shown as linearly arranged, clearly there is need for backtracking, particularly between the requirements gathering, analysis, and SRS preparation stages. Still, conceptually, there is precedence among them - before analysis is done the requirements must be gathered, and before they can be specified, they must be analyzed. Hence, in the process specification these are linearly organized. The activities in the process for this phase are given below. Most of these activities are self-explanatory. However, a brief description of the main activities is given later.

Prepare for requirements gathering and analysis
- Do background reading on technical / business concepts and undergo training
- Familiarize with customer's methodology and tools to be used
- Identify methods for information gathering
- Prepare questionnaires for eliciting information
- Identify user groups and interviewees
- Plan prototyping
- Define requirement specification standards
- Develop interview plan and review with customer

Gather requirements
- Establish objectives and scope of the system
- Gather functional requirements
  - Identify business events
  - Identify inputs and outputs for each business event
  - Determine relationship between inputs and outputs
  - Determine precedence relationships among events
- Gather information on external interfaces
- Gather operating environment requirements
- Gather performance requirements
- Gather standards requirements
- Gather special user requirements
- Prepare prototypes
- Evaluate prototypes
- Conduct feedback sessions (on current understanding of requirements)
To be able to better understand the requirements of the customer, background reading is required in the domain of the business system as well as in technical concepts relating to hardware and software platforms, language etc. which are relevant to the system being developed. If the customer follows a particular methodology or uses certain tools, familiarization with them is required. When standard information and answers to closed-ended questions are to be gathered from many users, questionnaires become very useful. Hence the analyst should prepare questionnaires relating to the customer's business and operating environment. The analyst should also identify the user groups and other people that need to be interviewed. The organization structure can help in identifying the interviewees. Usually, the time the users can devote to giving information and requirements is limited. Hence a plan for interviewing should be made to make effective use of the time available for requirements gathering, keeping in mind the order of information gathering and the availability of people.

A prototype is a limited simulation of the application to be developed, and is usually created to give the user a feel for what the application would look like once developed. If prototyping is needed in the project, then prototyping objectives have to be established in advance and development and feedback collection planned.

Gathering requirements involves obtaining all relevant information that will help in understanding the customer's requirements. Customer's requirements can be classified as business, functional, interface, operating environment, performance, standards, and special requirements. Business requirements establish the goals and scope of the system. Functional requirements are the end-users' view of required business functionality, and implementing them is the basic purpose of the project. The business system basically responds to and processes a set of events. Hence understanding the business functions, for the most part, involves understanding the various events that take place and the business processing that happens in response to these events. Events can be of two types – external events which are triggered by external entities, or temporal events which are triggered by time. Besides identifying the events, it is necessary to identify inputs and outputs for all events. The relationship between inputs and outputs of an event is really the processing that needs to be performed by the system in response to a business event.

The application that is being developed might interface with many other existing automated and non-automated systems. These external interfaces have to be identified to make sure that the application will pass and receive data compatible with the external interfaces. The hardware and software environment in which the application is expected to operate needs to be clearly understood, as they impact the design and implementation. The performance requirements are constraints on response time, throughput, etc., that are placed by the business. These requirements may be different for peak and normal conditions.

If some standards must be followed in the project, then such standards need to be identified, including standards for user interfaces, coding, documentation, etc. Special user requirements encompass all supporting requirements. They include safety, security, reliability, backup, transaction, legal requirements etc. They can have a major impact on the system and the effort required in building it. Hence, they must be clearly understood.

Prototypes give the users a feel of the system and are a useful technique in gathering requirements. If prototyping is to be used, the prototypes should be demonstrated to as many users as possible, and feedback collected.
The goal of analysis is to identify the requirements in a complete, accurate, consistent and unambiguous fashion from the information collected. Analysis accomplishes this by constructing models of the system. The models concentrate on describing what a system does rather than how it does it. The process model of a system is a representation of the processes that transform the data in the system.

The data model gives a logical structure to the user's view of the data in the system. Usually the data model is depicted in the form of an entity relationship. The entity relationship diagram shows the business entities in the system and the relationships between these entities. Entities can be thought of as the data groups that participate in the business processes. The physical database is later derived from the data model. A data dictionary is the central repository of all objects in the process model and data model. It describes the data stores, processes and external entities in data flow diagrams.

Once the analysis is done, the SRS can be prepared. This task involves documenting the objectives and scope of the system and consolidating the process model, data model, and data dictionary etc. into a document. Unresolved issues should also be included in this document. Acceptance criteria are the list the conditions under which the customer will accept the delivered software. Preparing the acceptance criteria explicitly helps have a clear understanding with the customer about what the customer considers acceptable software. It also helps in quantifying the customer's expectations from the software. A review of requirements specification document and the acceptance criteria should be done, after which a sign-off can be obtained from the customer.

The process described above is for traditional function-oriented analysis. For projects that want to follow an object-oriented approach, a separate process has been defined. For requirements, the process for an object-oriented approach is almost similar, except for the “analyze requirements” activity in the requirements process. In an object-oriented analysis, this is done using use cases, which is one of the standard ways of doing object-oriented analysis. In this approach, for each business event, a use case is written (instead of drawing a process diagram), generally in the form of narrative text. For the use cases, various actors (i.e. users) are identified, along with the interaction of the actors with the system. These interactions or dialogues form the use cases. From the use cases and other analysis, classes are identified along with some methods, and a class dictionary is formed.

Requirements Specifications
The planning, elicitation, and analysis activities culminate in requirement specification. The SRS document is the main output of the requirements stage. To ensure that all necessary requirements are specified, one simple approach is to specify a template for the SRS. The basic contents of the template used at Infosys are described below. The various items in the template also form a checklist to ensure completeness. The template is somewhat different if an object-oriented analysis is done.
1. **OVERVIEW**
An introduction to the system to be built.

1.1. **Current System**
A brief description of the current system, if a system exists.

1.2. **Limitations of the Current System**
A list the limitations of the current system.

1.3. **Proposed System**
An overview of the proposed system.

1.2.1. **Objectives of the Proposed System**
List the business objectives/expected benefits of the proposed system.

2. **FUNCTIONAL REQUIREMENTS**
List of requirements related to the customer's business.

2.1. **System Requirements**

2.1.1. **Scope & Boundary**

2.1.2. **Context Diagram**

2.2. **Business Events**

2.2.1. **External Events**
List of external events. External events are triggered by external entities.
E.g., a client calling in to place an order or a user entering a command.

2.2.2. **Temporal Events**
List of temporal events. Temporal events are triggered by time. E.g.
producing a summary report everyday at 9 p.m..

2.3. **Inputs and Outputs**
Give inputs and outputs for each business event.

2.4. **Relationships**
Specify relationship between inputs and outputs.
2.5 Precedence Relationships
Specify any precedence relationship between events.

2.6 Screens
2.7 Reports

3. EXTERNAL INTERFACE REQUIREMENTS
The application that is being developed might interface with many other existing automated and non-automated systems. These external interfaces have to be identified to make sure that the application will pass and receive data compatible with the external interfaces. They are specified here.

4 OPERATING ENVIRONMENT REQUIREMENTS
4.1 Hardware
4.2 Software
4.3 Network
4.4 Communication

5 PERFORMANCE REQUIREMENTS
All performance requirements are specified here. Examples are on-line response time, no. of transactions per second, no. of customers to be serviced per hour, constraints on batch job window etc.

6 STANDARDS REQUIREMENTS
All standards that the customer requires to be followed during the project should be listed here. The actual standards themselves can be defined in a separate document.

6.1 User Interface
6.2 Detailed Design
6.3 Coding
6.4 Document

7 SPECIAL USER REQUIREMENTS
7.1 Security
7.2 Audit Trail
7.3 Reliability
7.4 Transaction Volume and Data Volume
7.5 Backup and Recovery
7.6 Legal
7.7 Data Migration
7.8 Data Retention
7.9 Installation
7.10 User Training
7.11 User Manual and Help
7.12 Automated and Manual Functions
7.13 Features Not Required

8 CONSTRAINTS

9 PROTOTYPE
If a prototype exists or is to be built, reference should be given to it

10 GLOSSARY OF TERMS
Requirements Change Management

Requirement change management process defines the set of activities that need to be performed when there are some new requirements or changes to existing requirements (we will call both of these as changes in the requirements). Requirement changes can occur at any point during the project execution stage. The basic goal of requirement change management process is to control requirement changes and minimize the impact of changes on the project. This involves understanding the full impact of a requirement change request, as well as the cumulative impact of changes, on the project. It also requires making the customer fully aware of the impact of the changes on the project so that changes in the negotiated terms can be done amicably. The requirements change management process, in a sense, tries to ensure that a project succeeds despite requirement changes.

The Process

There are two aspects to requirements change management – agreement with the customer about how the changes will be dealt with, and the process of actually making the changes. The overall approach for handling changes has to be agreed by the customer, and is frequently a part of the proposal as well as the project management plan. Generally, this specifies how the change requests will be made, when formal approvals are needed, building a buffer in the estimates for handling changes etc. In the context of the overall approach, when a request for a requirement change comes in, the requirements change management process has to be executed.

The project leader is primarily responsible for executing the process to incorporate the change in the project. However, the customer, the business manager to whom the project leader reports, and the development team also participate in this process. The entry criterion for this process is that a change request has been received, and the inputs are the change request and the work products that have already been produced in the project. The main outputs are the impact analysis report for the change request, revised project plan, and changed work products, and the exit criterion is that the change has been incorporated. The major steps in the process are:

- Log changes
- Perform impact analysis on the work products
- Estimate effort needed for this change requests
- Re-estimate delivery schedule
- Perform cumulative cost impact analysis
- Review impact with senior management, if thresholds are exceeded
- Obtain customer sign off
- Rework work products

A change request log is maintained to keep track of the change requests. Each entry in the log contains a change request number, a brief description of the change, the impact of the change, the status of the change request and key dates. The effect of a change request is assessed by performing impact analysis. Impact analysis involves identifying work products that need to be changed, and evaluating the quantum of change to each; reassessing the project’s risks by revisiting the risk management plan; and evaluating the overall impact of the changes on the effort and schedule estimates. The outcome of the analysis is reviewed and approved by the project leader and the customer. The change request itself is incorporated in the requirement specification document, usually as appendices. Sometimes the relevant portions of the document may also be modified to reflect the changes. Monitoring of approved change requests and ensuring proper implementation of change request is handled by the configuration management process, which is discussed in a later chapter.

Examples

To specify the changes and the output of the change management process, a simple template has been defined, which contains summaries of various attributes. Each change is assigned a unique number for reference that is specified by the request number field. The change specification gives a brief description of the requested change. The category of change (e.g.
design change, contract change, functionality change, performance change, etc.) may also be specified. The nature of the change may also be specified as change category. The summary of the impact analysis is recorded, in which brief information is given regarding work products that will be impacted, the effort involved, and the impact on schedule. The state of the change request, i.e. what is being done with this request, is recorded in the status field. The date of change request may also recorded, along with the date the change was approved, if approval is needed.

Two examples of change requests are given below, which use a customized version of the change request template. In these examples, the detailed contents of the impact analysis are not important for the purposes of understanding requirements change management. Though the change request is specified using the template, the actual tracking of implementation of a change request is handled by the configuration management process, which is discussed later.

<table>
<thead>
<tr>
<th>Project</th>
<th>XYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req. No.</td>
<td>10</td>
</tr>
<tr>
<td>Date: 23 Feb 1998</td>
<td></td>
</tr>
<tr>
<td>Change Spec.</td>
<td>PFNETCONFIG – Packed format netconfig support</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>CDMA needs to parse three new tables – CDMACONF, CDMACELL and SBSINV. Two new modules need to be added:</td>
</tr>
<tr>
<td></td>
<td>1) DumpTab Parser module</td>
</tr>
<tr>
<td></td>
<td>2) Cdmmapnetconfig module: Uses DumpTab parsers to extract table objects, then use resource objects to save configuration data.</td>
</tr>
<tr>
<td>on Schedule</td>
<td>Nil</td>
</tr>
<tr>
<td>on Effort</td>
<td>10 person-days</td>
</tr>
<tr>
<td>Status</td>
<td>Will be incorporated in the new CDMA package.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>XYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req. No.</td>
<td>11</td>
</tr>
<tr>
<td>Date: 23 Feb 1998</td>
<td></td>
</tr>
<tr>
<td>Change Spec.</td>
<td>IS41 Analyzer - IS-41 Analyzer support for CDMA</td>
</tr>
<tr>
<td>Impact Analysis</td>
<td>No particular change in configuration module and analyzers for CDMA. The TDMA code can be reused as such. Scripts can also be reused. Netconfig and analyzer classes can be reused. The impacted modules are:</td>
</tr>
<tr>
<td></td>
<td>1) cgaapp module: Has to trigger analysis for IS-41 also, separately.</td>
</tr>
<tr>
<td></td>
<td>2) cdmaroi module: (a) TRIS41ROI has to be copied as TRCDMAIS41ROI. (b) There is a pure virtual method in TRCDMAROI for setting the ActualCallModelManager. This needs to be redefined.</td>
</tr>
<tr>
<td></td>
<td>3) silver06guiapp++ module: IS-41 has to be added in the resourceList.</td>
</tr>
<tr>
<td>on Schedule</td>
<td>Nil</td>
</tr>
<tr>
<td>on Effort</td>
<td>5</td>
</tr>
<tr>
<td>Status</td>
<td>Will be incorporated in the new CDMA package.</td>
</tr>
</tbody>
</table>