* + 1. **Modelling Requirements**

**We build models in requirements analysis to understand**

* current systems or business processes which we are trying to automate
* how users will use a new system

The **software requirements** document is the official statement of what is required of the system developers.

* Should include both a definition of user requirements and a specification of the system requirements.
* It is NOT a design document. As far as possible, it should set WHAT the system should do rather than HOW it should do it.

**Requirements Document Variability**

Information in requirements document depends on the type of system and the approach to development used.

Systems developed incrementally will, typically, have less detail in the requirements document.

Requirements documents standards have been designed e.g. IEEE standard. These are mostly applicable to the requirements for large systems projects.

**The Structure of a Requirements Document**

|  |  |
| --- | --- |
| **Chapter** | **Description** |
| Preface | This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version. |
| Introduction | This should describe the need for the system. It should briefly describe the system’s functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software. |
| Glossary | This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader. |
| User requirements definition | Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified. |
| System architecture | This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted. |

|  |  |
| --- | --- |
| System requirements specification | This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined. |
| System models | This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models. |
| System evolution | This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system. |
| Appendices | These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data. |
| Index | Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on. |