

Software Requirements

- Dataflow model includes input, output and processing only
- In dataflow model, decomposition will be based on functionalities or “**what function has to be done**”. But in object-oriented model, decomposition is based on “**who is doing the function**”.
- Reuse occurs at every level of software development. Occurs both at design as well as documentation level.
- After developed many software, (90 to 92%) of software code is present in repositories already, and only 10% of the code to be written. (Reuse of already made code).

Requirements analysis and specification

- In requirements analysis, you are not making any kind of implementation analysis.
- **SRS documents**: Software requirements specification. Is a document made before starting the actual design of the software. It includes all the requirements by the client.
- After SRS has been made, design process will transform it to the documentation.
- **Acceptance-testing** is done on customer side.
- There are two kinds of levels: **high-level design and low-level design**. High level design includes the whole software development lookup. While low level design includes only a particular module development.

Low-level design includes module specification (MSPEC). MSPEC includes data structures and algorithms used in the module.

Three important components in modular structure based development are set of modules, control relationship between them and interface between them.

- At the top are the manager modules and at the bottom of the tree, are the core modules. This is called structure chart of the software process. Very much effort to be placed so that they can be represented as a tree like structure.
- To maintain strong relationship, pass by reference and object passing is generally avoided in between the modules. Interfaces must be simple.



The main advantage of using modular structures or modularity in object-oriented design is that updates that comes to the software are essentially the addition of new modules to the whole project code. (Low-coupling ensured) Hence, our first task is to make them modularized.

Functional and Non-functional Requirements

- **Functional Requirements :**
 - Functional requirements are represented or stated in the form of input to be given to the system, the operation performed, and the output expected. They are basically the requirements stated by the user which one can see directly in the final product.
- **Non-functional requirements:** They are also called non-behavioral requirements.
 - It includes :
 - Portability
 - Security
 - Maintainability
 - Reliability
 - Scalability
 - Performance
 - Reusability
 - Flexibility

Aspect-oriented design focuses on separation of concerns, which is very important for building good software. (**Concerns means problems and in object-oriented design, problems are objects itself, so we need to separate the objects**).
