Date / /
Page

29/08/17 Module - 3

The process of understanding the real world problem & users need and exposing abstract soln to problems is known as problem analysis.

- With the help of problem analysis one can gain a better understanding b4 ~~b~~ore actual development begins of the problem to be solved.

- ~~It avoids~~
- There are 5 steps of problem analysis :-

(i) Gain agreement on the problem definition:

(ii)
- Write a simple & clear definition.
- Establish an order of importance for all features of the system.
- Come to an agreement with all stake holder.
- Resolve conflicts by negotiating.

(ii) Identify the root causes of the problem:-
- Make sure the problem identified is a real problem.
- Sometimes a problem hides ~~out~~ other more important problems.
- Addressing the wrong problem may lead to failures.
- A problem can have several causes.
- This part of analysis requires input from extremly knowledgeable & insightful & experienced persons.

(iii) Identify stake holders & users :-
- Stake holder :- Anyone who could be affected by the new system or has input to provide in the implementation of the new system.

*) Complex problems always involve the input of diff stake holders that have diff. view points on the problem.
*) Forgetting one of these stake holders might lead to major rework of even project failure.

Software requirements specifications (SRS)

It is a description of software system to be developed. It lays out functional and non-functional requirements and it may also include a set of used cases that describe user interactions that software must provide.

→ It establishes the basis for an agreement b/w customers and contractors or suppliers.

→ It permits a rigorous assesment of requirements before design can begin & reduce later redesign.

*) Rolee of SRS :-

The SRS is a communication tool b/w stake holders and software designers.

1) Facilitating reviews.
2) Describing the scope of work.
3) Providing a reference to software designers.
4) Providing a framework for testing primary and
5) secondary used cases.
5) Including features to ensure requirements.

Nimra Kaka ♡ ♡ ♡ HIHO KAKA

*) Software Prototyping :-
→ It is the activity of creating prototypes of software applications.
→ Prototype is the incomplete version of the software program being developed.
→ A prototype may be completely different from the final product and only stimulates a few aspects of the product.

*) Benefits of Prototyping :-
→ The software designer can get valuable feedback from the users early in the project.
→ The client and the contractor can compare if the software made matches the software specifications.
→ It also help software engineer to estimate deadlines and mile stones in the project development.

*) Types of prototyping :-

1). Throwaway prototyping.
2). Evolutionary prototyping.
3). ~~Inter~~ Incremental prototyping.
4). Extreme prototyping.

Throwaway prototype :- [Close ended prototyping]
Rapid prototyping :- refers to the creation of a model that will eventually be discarded rather than becoming part of the final delievered software.

Evolutionary prototyping (Read board prototyping) :-
The main goal of evolutionary prototyping

(flexible)
is to build a robust type in structural manner & constantly refinning.

#) Incremental prototyping :-

The final product is built as a separate prototype. At the end the seperate prototypes are merged in an overall design which helps in the reduction of time gap b/w user & software.

#) Extreme prototyping :- It is mostly used in web application development where web development is divided into three phases :-

1). Static prototyping in which HTML pages are developed.
2) The screens are programmed and functionality is added using a simulated service layer.
3). The services are implemented.

Advantages & Disadvantages same as prototype model.

#) Cohesion Measure & coupling :-

Coupling → An indication of the strength of interconnection b/w program units highly coupled applications have program units dependent on each other & loocely coupled or independent.

Modules :- They are independent.

Cohesion :→ Its a measure of how well modules fit together.

→ should be more for good application

*) Coupling	Cohesion.
→ It is the indication of relationship b/w modules.	It is the indication of relationship within modules.
→ It shows relative independence among modules.	It shows modules relative functional strength.
21/09 → It is a degree to which a module is connected to other modules	Cohesion is a degree to which a component (module) focuses on a single thing.
→ Making private field, private methods, and non-public classes provide loose coupling.	Cohesion is the kind of natural extension of data hiding. Ex: class having all members visible with a package having default visibility.
→ Inter module	Intra module.

Structure design :→
is a conceptualization of problem into several well organised elements of solution. It is concerned with the solution design and gives a better understanding of how the problem is being solved.

- It is mostly based on divide & conquer strategies. Where problem is broken into several small problems & each problem is solved individually until the whole problem is solved.

- A good structure design has high cohesion & low coupling.

Function oriented design :-

Tutorial point.

There are two generic approaches for software designing :-

1) Top down design.
2) Bottom up design.

Top down design takes the whole system as one entity and then decomposes it to achieve more than one sub-system.
Each sub-system is then treated as system & then decompose further. This process keeps on running until the lowest level of system is achieved.

Bottom up design
This model starts with most specific & basic components & proceeds with composing higher level of components using basic components.
It keeps creating higher level component until the desired system has not evolved as one single component.