TNPL பக்கம் : 23
நாள் : / /

Software Cost Estimation:

There are many factor to be considered when estimating the total cost of a project.

This include labour, hardware and Software purchase or rental travel for meeting or testing purpose, tele communication, video conferences, training courses, office spaces so on.

Steps for Estimation:-

- Estimate the size of development product.

This generally ends up either line of code or functional point (FP).

- Estimate the effort in person month or person hour.
- Estimate the schedule in calender month.
- Estimate The project cost

in dollars or local currencies.

Project Estimation Guidelines:-

* Delay Estimation until late in the project.
* Base estimates on similar project that have already been completed.
* Use relatively simple decompose technique to generate project cost and effort estimates.
* Use one or more empirical model for software cost and effort estimation.

Software Cost Estimation Techniques:-

Cost estimates can be made either top down or bottom up.

Top down estimation focuses on system level cost such as computing resources and personal required to develope the system,

cost of Configuration management, Quality assurance, System integration, training and publication.

Bottom up cost estimation estimate the cost to develope each module or Sub System.

There are four widely used techniques. They are

1. Expert Judgement
2. Delphi cost Estimation
3. Work breakdown Structure
4. Algorithmic cost model

Expert Judgement:-

It is a top down estimation technique, it involves Consulting one or more expert. The expert provide estimates using There own method and experiences.

The expert must Confident

That project is similar to previous one, but may have overlooked some factor that make a new project significantly different.

DELPHI COST ESTIMATION :-

It involves more interaction and communication between those who are participating.

The procedure are as follows:-

* The coordinator presents a specification and estimation to each expert.
* coordinator calls a group meeting in which expert discuss estimation issues, with the coordinator and each other.
* Each ~~Coordinator~~ Experts fills out forms unanimously.
* Coordinator prepare and distribute the summary of the estimates.
* The coordinator then call a

group meeting. In this meeting the experts mainly discuss the points where there estimates widely vary widely.

* The experts once again fill out the forms unanimously.

* Again coordinator satisfied with the Overall prediction Synthesised from the prediction.

The key to this Success is expert coordinating, the coordinating must be talented enough to synthesis the diverse and wide ranging statement.

After The estimation session, project manager Summarize the results and reviews with the team at a Software project.

Work Breakdown Structure (WBS):

It is a hierarchical chart that accounts for the individual part of a system.

WBS chart can indicate either a product hierarchy or process hierarchy.

Product hierarchy identifies the product comment and indicate manner in which components are interconnected.

Process hierarchy identify the work activities and relationship among these activities.

Using WBS techniques cost are estimated by assigning cost to each individual component in a chart.

The primary advantages of WBS techniques are in identifying and accounting for various product.

Various product and process factors and making a exactly

which cost are included in estimating of a software cost.

Algorithm Cost Model:-

It is a bottom up estimator. It is based on mathematical ~~method~~ model that produce cost estimate as a function of number of variable, which are considered to be the cost factor.

The Constructive cost model (COCOMO) is an algorithmic cost model developed by berry boehm. The model uses a basic recursion formula with parameter that are derived from historical project data and current project characteristics.

Software Maintenance:-

It is an activity in which the program is modified after it has been put in use.
Maintenance is a process in which changes implemented by either modifying existing system architecture or by adding a new components to the system.

Types of Software maintenance:-

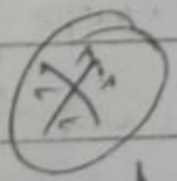
* Corrective maintenance:-
  Correction of Software faults.
* Adoptive maintenance:-
  Adopt the change in Environment.
* Perfective maintenance:-
  Modifying or enhancing the System to meet the new requirement.
* Perfective maintenance:-
  changes made to improve future maintainability.

Estimating Software Maintenance Cost:-

Software maintenance require 40 to 60% and in some cases as much as 90% of total life cycle effort devoted to software product.

Maintenance activity include adding enhancement to product adopting to product to new processing environment and correction problem.

A widely used rule for distribution of maintenance activity is a 50% for enhancement 20% for adoptation and 20% error correction.



- Software Characteristics
- S/W maintenance
- S/W matrix metrics measurement
- Prototype module
- Delphi method
- Software feature.

chap 6: SRS

TNPL UNIT NO: 17 PAGE NO: 1 1

Software requirement Specification (SRS)

It is a set of document that contains a clear specification of functional, performances, design and interface requirement of a proposed system.

It is a complete description of the behaviour of the system to be developed. It include set of used cases that describes all of the interaction that the user will have with the software.

Use cases are also known as functional requirements

- It should specify only external system behavior.
- It should specify constraint on the implementation.
- It should be easy to change.
- It should serve as a reference tool for system maintenance.
- It should characterized acceptable responses to undecide event.

* characteristics of SRS:

* Correctness
* Completeness
* Unambignous
* Verifiable
* Consistant
* Modifiable
* Traceable

chap 8

Objective of a Software design:-

* Identify different types of a

Source formal

Specification Techniques :-

(1) Relational Notations

It is based on entities and attributes.

(a) Implicit equations

State the properties of a solution without stating a solution method.

(b) Recurrence relations

It consist of an initial part called the basis & one or more recursive parts.

(c) algebraic axioms

Used to specify the properties of abstract data types.

(d) Regular expression

It is brief notation for defining both finite/infinite sets of symbol strings.

(2) State-oriented Notation

(a) Decision table used to specify actions in terms of complex decision criteria.

(b) Event Tables specify actions to be taken when events occur under different set of conditions

(c) Transition tables specify changes in the state of a system as a function of driving forces.

(d) Finite state Mechanism DFD, regular expression transition table are combined to provide a powerful finite state mechanism for functional specification of the system

Source formal

Specification Techniques :-

(1) Relational Notations

It is based on entities and attributes.

(a) Implicit equations

State the properties of a solution without stating a solution method.

(b) Recurrence relations

It consist of an initial part called the basis & one or more recursive parts.

(c) algebraic axioms

Used to specify the properties of abstract data types.

(d) Regular expression

It is brief notation for defining both finite/infinite sets of symbol strings.

(2) State-oriented Notation

(a) Decision table used to specify actions in terms of complex decision criteria.

(b) Event Tables specify actions to be taken when events occur under different set of conditions

(c) Transition tables specify changes in the state of a system as a function of driving forces.

(d) Finite state Mechanism DFD, regular expression transition table are combined to provide a powerful finite state mechanism for functional specification of the system

TDEV=2.5*(44)**0.32=8.4

Effort multipliers are used to adjust the estimate for off-nominal aspects of the project.

3.4 Staffing-Level Estimation

The number of personnel required throughout a software development project varies at every stage

For example,

- Planning and analysis are performed by a small group of people.
- Architectural design by a large group of people and the detailed design by a larger number of people.
- Implementation and testing required the largest number of people.
- The early phase of maintenance may required numerous personnel, but the number should decrease in a short time

In 1958, Norden observed that research and development of project follow a cycle of planning, design, prototype, development and uses, with the corresponding personnel utilization shown in the following figure 3.5

Figure 4.5 – A Conflict Situation

In the above illustration, both t_1 and t_2 are enabled only are con fire firing one will disable the other.

4.5 Language and processors for requirement specification

A number of special purpose language and processor have been developed to permit concise statement and automated analysis of requirements specification for software.

Most of the specification languages are graphical in nature, while others are textual; all are relational in nature.

4.5.1 PSL/ PSA

PSL stands for problem statement language. The Problem Statement Analyzer (PSA) is the processor PSL/PSA ~~been or~~ originally developed for data processing application widely used in other application.

The objective of PSL is to permit expression of much of the information that commonly appears in SRS.

In PSL, system description can be divided into eight major aspects,

1. System input/output flow.
2. System structure
3. Data structure
4. Data derivation
5. System size and volume
6. System dynamics
7. System properties.
8. Project management

processing requirements.

The PSA system can provide report in four categories: Data base modification report, reference reports, summary reports and analysis reports.

PSL/PSA has been used in many different situations, ranging from commercial data processing applications to air defense system.

4.5.2 RSL/REVS

RSL stands for requirement statement language REVS stands for requirement engineering validation system which processes and analyzes RSL statement.

Both RSL and REVS are comports of s/w requirement Engineering methodology many of the concepts in RSL are based on PSL the fundamental characteristics of RSL is flow oriented approach used to describe red time systems.

The Requirement Engineering and validation system (REVS) operates on RSL statement REVS consists of major components:

1. A translator for RSL
2. A centralized database, the abstract semantic model (ASSM)
3. A set of automated tools for processing information in ASSM

4.5.3 SADT

SADT stands for Structured Analysis and Design Technique. SADT incorporates a graphical language and asset of methods and management guidelines for using the language

The SADT language is called the Language of structured Analysis (SA). SADT is the Interconnection Structure of any large, complex system.

The SA language and the procedure for using it are similar to the engineering blue print system used in civil and mechanical engineering.

An SADT model consists of an ordered set of SA diagrams. Each diagram is drawn on a single page, and each diagram must contain 3 to 6 nodes plus interconnecting are.

Two basic types of SA diagrams are the activist diagram (act gram) and the data diagram (datagram)

On an act diagram the nodes denote activities and they are specifies data flow between the activates.

Following figure 4.7a illustrates the formats of activity data diagram and datagram nodes.

The SADT can be applied to all type of systems; it is not limited to s/w applications. On the other hand, one would probably use SADT only an large compiled projects.

4.5.4 Structured system Analysis (SSA)

Two similar versions of structured system Analysis, have been described by Gane and Sarson; and by Demarco

(i) Gane and Saron version used in data processing applications that have data base requirements.

(ii) Demarco version suited to data flow analysis SSA is primarily used in traditional data processing environments like SADT, SSA also uses a graphical language to build models of system, SSA data flow diagrams are similar to SADT actigrams but they do not indicate mechanism and control.

4.5.5 Gist

Gist is a formal specification language.gist is a textual language based on a relational model of objects and ~~elutriates~~. attributes

A Gist specification is a formal description of valid behaviors of a system

A specification is composed of three parts:

1. A specification of object types and relationship between these types. This determines a set of possible states.
2. A specification between possible states
3. A specification of constraints on states and state transitions.

Gist has a well- defined syntax learning the syntax of gist is similar to learning the syntax to learning the syntax of a new programming language.

Learning to use Gist is complicated by the fact that one is not only learning a new notation but also learning a new way of thinking about systems and learning new techniques for specifying functional behavior.

Summary

- Software requirement definition is concerned with preparation o