Sct 221-c004-0518/2022

Software engineering

Project

A project is a collection of tasks that must be finished to achieve a specific goal. A project may be defined as a collection of inputs and outputs needed to accomplish a particular objective.

Project management

-project management is a discipline and skill for organizing and managing software projects. It is a branch of software project management that deals with the planning, execution, supervision, and control of software projects.

E.g. For the purpose of creating computer software that satisfies criteria, a process of managing, assigning, and timing resources is used.

The client and the developers must be aware of the project's duration, budget, and schedule while managing software projects.

Cost estimation models

Software products are said to be feasible if they are developed within the budget constraints. Prior to making software product it’s imperative to predict the software development cost.

Practitioners have expressed concern over their inability to accurately estimate costs associated with software development. This concern has become even more pressing as costs associated with development continue to increase. As a result, considerable research attention is now directed at gaining a better understanding of evaluating software cost estimating tools.

software cost estimation models examples: COCOMO II, COCOMO, PUTNAM, STEER and ESTIMACS based on the parameters implement ability, extensibility, flexibility and traceability and techniques used to estimate software costs.

Software cost estimation model is an indirect measure, which is used by software personnel to predict the cost of a project.

They are used for the number of purposes, whuch include:

• Budgeting Overall estimate has to be accurate, the most desired capability. Hence initial efforts are directed in predicting budget for the software product.

• Tradeoff and risk analysis An important additional capability is to illuminate the cost and schedule sensitivities of software project decisions (scoping, staffing, tools, reuse, etc.).

• Project planning and control An additional potential is to provide cost and schedule breakdowns by component, stage and activity.

• Software improvement investment analysis Strategies such as tools, reuse, and process maturity benefit the development process of software.

COST ESTIMATION MODELS

-The fast changing nature of software development has made it very difficult to develop parametric models that yield high accuracy for software development in all domains.

S/w development costs hikes abnormally and practitioners continually express reckon over their incapability to accurately predict the costs involved.

S/w models constructively explain the development life-cycle and accurately predict the cost of developing a software product.

Many s/w estimation models have evolved in the last two decades based on the pioneering efforts by the researchers. Mostly being proprietary models cannot be compared and contrasted as far as the model structure is concerned .

Theory or experimentation determines the functional form of these models. These are:

1. COCOMO 81

1(a) Basic COCOMO

COCOMO is an acronym used for Constructive Cost Model. It was first published in 1981 book Software Engineering Economics by Barry Boehm.

It gives the magnitude of cost of project due to the ease of openness of model. It is meant for relatively small projects as a very few cost drivers are associated with it.

Its supportive when the team size is small, i.e. small staff. It’s good for quick, early, rough, order of magnitude of software costs, but its accuracy is necessarily limited because of its lack of factors to account for difference in hardware constraints, personnel quality and experience, use of modern tools and techniques and other project attributes are known to design) of the software engineering process.

1. COCOMO-II

The COCOMO II research effort was started in 1994 at USC. Its major focus on non-sequential and rapid development process models, reengineering, reuse driven approaches, object oriented approaches, etc. It is a cumulative result of three variants, Application composition model, Early design model, and Post architecture model.

The Application Composition model is worn to approximate effort and schedule on projects that use Integrated Computer Aided Software Engineering tools for rapid application development. It is based on Object Points (Object Points are a tally of the screens, reports and 3 GL language modules developed in the application).

. The Early Design Model involves the investigation of substitute system architectures and concepts of operation.

. The Post-Architecture Model is used when apex level design is complete and thorough information about the project is accessible and as the name suggests, the software architecture is sound defined and well-known. It accounts for the intact development life-cycle and is a exhaustive extension of the Early-Design model. This is a lean-to intermediate COCOMO model and defined as:- EFFORT = 2.9 (KLOC)1.10

1. PUTNAM MODEL (SLIM)

SLIM (Software Life Cycle Model) is based on Putnam’s study in terms of Rayleigh distribution of project personnel level versus time. It chains most of the popular size estimating methods including ballpark techniques, source instructions, function points, etc.

It estimates project effort, schedule and defect rate. Record and analyze data from formerly completed projects which are then used to standardize the model

If data are not obtainable then a set of questions can be answered to get values of MBI and PF from the presented database. Productivity, P, is the ratio of software product size S and development effort E is that is P= The Rayleigh curve [2] is accustomed to define the distribution of effort which is modeled by the differential Equation.