**OBSERVATIONS ON ESTIMATING**

Estimation of resources, cost, and schedule for a software engineering effort requires experience, access to good historical information, and the courage to commit to quantitative predictions when qualitative information is all that exists. Estimation carries inherent risk and this risk leads to **uncertainty**.

* *Project complexity*
* *Project size*
* The *degree of structural uncertainty*

**PROJECT PLANNING OBJECTIVES**

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempt to define best case and worst case scenarios so that project outcomes can be bounded.

**SOFTWARE SCOPE**

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering

*Software scope* describes the data and control to be processed, function, performance, constraints, interfaces, and reliability.

**Obtaining Information Necessary for Scope**

The first set of context-free questions focuses on the customer, the overall goals and benefits and that will lead to a basic understanding of the problem. For example, the analyst might ask:

• Who is behind the request for this work?

• Who will use the solution?

• What will be the economic benefit of a successful solution?

• Is there another source for the solution?

The next set of questions enables the analyst to gain a better understanding of the problem and the customer to voice any perceptions about a solution:

• How would you (the customer) characterize "good" output that would be generated by a successful solution?

• What problem(s) will this solution address?

• Can you show me (or describe) the environment in which the solution will be used?

• Will any special performance issues or constraints affect the way the solution is approached?

The final set of questions focuses on the effectiveness of the meeting. Gause and Weinberg call these "meta-questions" and propose the following (abbreviated) list:

• Are you the right person to answer these questions? Are answers "official"?

• Are my questions relevant to the problem that you have?

• Am I asking too many questions?

• Can anyone else provide additional information?

• Should I be asking you anything else?

What is the primary source of information for determining scope?



**What issues should we consider when we plan to reuse existing software components?**

The following guidelines should be considered by the software planner when

reusable components are specified as a resource:

1. If off-the-shelf components meet project requirements, acquire them. The cost for acquisition and integration of off-the-shelf components will almost always be less than the cost to develop equivalent software. In addition, risk is relatively low.

**2.** If full-experience components are available, the risks associated with modification and integration are generally acceptable. The project plan should reflect the use of these components.

**3.** If partial-experience components are available, their use for the current project must be analyzed.

**How do I compute the expected value for software size?**

A three-point or expected value can then be computed. The *expected value* for the estimation variable (size), *S,* can be computed as a weighted average of the optimistic (*s*opt), most likely (*s*m), and pessimistic (*s*pess) estimates. For example,

*S =* (*s*opt *+ 4s*m *+ s*pess)/6