1. Important Points
   1. “Software project management encompasses the following activities: measurement, project estimating, risk analysis, scheduling, tracking, and control.” [15]
   2. “To be most effective, software metrics should be collected for both the process and the product.” [16]
   3. “All software project estimation techniques require that the project have a bounded scope and all rely on a high-level functional, decomposition of the project and an assessment of project difficult and complexity.” [16]
   4. “In order to ‘make provision against it,’ a software project team must conduct risk analysis explicitly.” [16]
   5. “Control focuses on two major issues: quality and change.” [17]
   6. “Software quality is defined as conforming to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.” [17]
   7. “Metrics for software quality … assess software from three distinct points of view: using it…, changing it…, and porting it.” [17]
   8. “Software quality assurance actually begins with the set of technical methods and tools that help the analyst to achieve a high-quality specification and the designer to develop a high-quality design.” [17]
   9. “Confusion arises when changes are not analyzed before they are made, recorded before they are implemented, reported to those who should be aware that they have occurred, or controlled in a manner that will improve quality and reduce error.” [18]
   10. “During its relatively short history, the creation of computer programs has evolved from an art form, to a craft, to an engineering discipline.” [19]
   11. “The first step in any software engineering project is customer communication.” [19]
   12. “The work product that evolves as a consequence of the application of an analysis method is the analysis model. The analysis model contains four generic elements: scenario-based elements…, class-based elements…, behavioral elements…, and flow-oriented elements.” [20]
   13. “[Software design methods] all rely on a set of fundamental principles: data and the algorithms that manipulate data should be created as a set of interrelated abstractions…, the internal design detail of data structures and algorithms should be hidden from other software components that make use of the data structures and algorithms…, modules should exhibit independence…, and algorithms should be designed using a constrained set of logical constructs.” [21]
   14. “[In software testing] our objective is to design tests that systematically uncover different classes of errors and to do so with a minimum amount of time and effort.” [22]
   15. “[Test case design methods] provide the developer with a systematic approach to testing. More importantly, methods provide a mechanism that can help ensure the completeness of tests and provide the highest likelihood for uncovering errors in software.” [23]
   16. “A software engineering pattern provides us with a template – a consistent method for describing an element or characteristic of the software process or the product that is produced by the process.” [23]
   17. “[Process patterns] enable a software engineering organization to develop a hierarchical process description that begins at a high level of abstraction.” [24]
   18. “A design pattern describes a design structure that solves a particular design problem within a specific context and amid ‘forces’ that may have an impact on the manner in which the pattern is applied and used.” [24]
   19. “We must build computer software faster.” [25] “To meet this challenge, software must be constructed from reusable components.” [25]
   20. “[Aging software plants] must be rebuilt and that demands a reengineering strategy.” [25]
   21. “To achieve [reuse and reengineering] we need… a new generation of software tools.” [26]
2. Disagreements

“Once [defining and negotiation the basic requirements for the software project] has been accomplished, requirements analysis begins. Two options are available at this stage: (1) the creation of a prototype… and/or (2) the creation of a detailed set of analysis models that describe the data, function and behavior of the system.” [19]

I disagree with the author’s assertion that after basic requirements for a project have been accomplished that only two options are available to a software engineer as the next step. I believe the next appropriate step depends on many factors, including: what software development model is being used, who your stakeholders are, the size and nature of the application, and the experience levels of your developers. Limiting yourself to a prototype or to analysis models as your only options at this stage is short sighted.

1. Questions and Clarifications

I do not have any questions about the material covered in this summary.