David Lambertson

**Software Maintenance: A Tutorial Summary**

**Important Points from Article**

* “The aim of the tutorial is to focus on solutions, not problems, but an appreciation of the problems in software maintenance is important. The solutions are categorized in a three-layer model: organizational issues, process issues, and technical issues.” [455]
* “It is helpful to understand trends and objectives of the wider field in order to explain the detailed problems and solutions concerned with maintenance.” [455]
* “A root problem for many software systems, which causes some of the most difficult problems for software maintenance, is complexity. Sometimes, this arises because a system is migrated from hardware to software in order to gain the additional functionality that is easy to achieve in software.” [456]
* “Once software has been initially produced, it then passes into the maintenance phase.” [456]
* “The inability to undertake maintenance quickly, safely, and cheaply means that for many organizations, a substantial applications backlog builds up.” [456]
* “In the 1950s, a large U.S. bank was about to take the major step of employing its very first full-time programmer. Management raised the issue of what would happen to this person once the programs had been written. The same bank now has several buildings full of data processing staff.” [457]
* “In the manufacture of a consumer durable, the majority of the cost lies in production, and it is well understood that design faults can be hugely expensive.” [458]
* “One of the problems of software maintenance is that it is very difficult to assess a software product to determine how easy it is to change.” [458]
* “Initial software development is product oriented; the aim is to deliver an artifact within budget and on time. In contrast, software maintenance is much closer to a service.” [459]
* “The foundation of good practice is a mature process, and the Software Engineering Institute at Carnegie-Mellon University has pioneered the development of a scale by which process maturity may be measured.” [460]
* “First of all, feasibility analysis is undertaken, in which the impact of the modification is asses, alternative solutions investigated, short- and long-term costs assessed, and the value of the benefit of making the change computed. Once a particular approach has been selected then the second stage of detailed analysis is undertaken. This determines firm requirements of the modification, identifies the software involved, and requires a test strategy and an implementation plan to be produced.” [461]
* “For example, configuration management and version control are indispensable for both. Information relating to development and maintenance will be kept in a repository.” [462]
* One of the major difficulties of software maintenance that encourages maintainers to be very cautious by nature is that a change made at one place in the system may have a ripple effect elsewhere, so consequently changes must be made.” [463]
* “In some cases, discarding the software and starting again may be the courageous, if expensive, solution, following analysis of the business need and direction, and the state of the software.” [464]
* “two important types of reverse engineering are redocumentation, which is the creation or revision of a semantically equivalent representation within the same relative abstraction layer, and design recovery, which involves identifying meaningful higher-level abstractions beyond those obtained directly by examining the system itself.” [465]
* “In safety-critical systems, for example, enormous effort is expended in producing and validating software.” [467]
* “We have described a three-level approach to considering software maintenance in terms of the impact on the organization, on the process, and on technology supporting that process. This has provided a framework with which to consider maintenance.” [468]

**Things I Didn't Agree With**

“Indeed, it may be said that any program that is sufficiently small to fit into a textbook or the understood by one person does not have maintenance problems.” [458]

I do not agree with this statement because there is some code out there small enough to fit into a textbook which can require maintenance. An example I have is that of an API connection. Some API calls can be done in 15-20 lines of code but will need maintenance. The API could change or the endpoint may need to be changed. This may be small maintenance, but it is maintenance nonetheless.

**Things I Did Not Understand**

I understood this article.