Written Assignment 6

Student

Univeristy of the People

**Question 1: Using *The Mythical Man Month reading resource located on the course homepage,* read the project problems mentioned in the text. List the difficulties and errors that occurred in these failed programming projects.**

“The Mythical Man Month” (Brooks, 1995) covers several failed large software development projects . The theme in the book is that large projects have have an intrinsic need for, which jeopardizes the integrity of the software.

One problem that large projects face is poor estimates of the time required to complete the project because of the use of the “man-month” to estimate the amount of time needed to complete a project. Brook’s (1995) argument is that the staff and month are only interchangeable if there is no communication needed between staff. However, as the number of people involved increases, so does the amount of time communicating and coordinating efforts. This problem is even more pronounced if individual parts cannot be completed simultaneously.

One of the examples that Brooks draws on to demonstrate the increased need for communication as more staff is involved comes from the Genesis account of the tower of Babel. He states, “ lack of communication led to disputes, bad feelings, and group jealousies. Shortly the clans began to move apart, preferring isolation to wrangling” (Brooks, 1995).

In order to facilitate communication in large projects, Brooks (1995) suggests that communication needs to be done informally, in meetings, and in a formal project workbook that documents every requirement and/or change in specification from the beginning of the project. He also mentions a proposal, by Parnas of Carnegie-Mellon University, to shield the programmer from details of the software besides the parts he is working on. I think that this is indeed the direction that programming has taken in order to reduce the problems associated with increasing the number of staff needed on a project.

**Question 2: Explain why the intangibility of software systems poses special problems for software project management?**

Project managers are responsible for planning the project as well as risk identification and risk management as the project proceeds. But project management for software projects is different from other projects because software is an intangible product. According to Sommerville (2011), software project management is more difficult than for other types of projects because of the intangible nature of the product. For one thing, the managers cannot judge the progress of the project by just looking at the status of the product during construction. When a bridge or an edifice is being build, a project manager can visually assess whether or not the project is progressing as planned. If the project schedule is slipping, it is easy to see that a part of the bridge or edifice has not be erected or completed. However, in the case of software project management, there is no concrete object to look upon to assess how the project is advancing (Sommerville, 2011). Instead, since software is not a tangible object, the manager has to rely on progress reports from the people developing the product. This means that the software project manager has to rely on other people instead of their own eyes to gather information about how the project is progressing. This puts software project managers at a disadvantage compared to other types of project managers whose product can be seen.

**Question 3: Why do the costs of assuring dependability increase exponentially as the reliability requirement increases?**

According to Sommerville (2011), the dependability of a software system, which impacts the user’s confidence that the system will work as expected, is its most important quality of the system. A software system’s dependability is affected by its availability, reliability, safety, and security (Sommerville, 2011). Unfortunately, the cost of assuring dependability tends to increase on an exponential scale the more reliable the system needs to be. There are two main reasons for this:

First, in order to produce higher levels of dependability, expensive hardware and development techniques must be implemented (Sommerville, 2011). One way to increase system dependability is to include redundant code or code that monitors the system for errors. Besides the additional cost of adding code that checks the system, the system performance can be affected by the inclusion of extra code. Often developers have to weigh the increased dependability of a system against the reduction in performance, and vice versa.

Second, convincing clients and regulators that the system has acceptable levels of dependability means increased testing and system validation. Sommerville (2011) provides us with the example of aircraft systems, which must demonstrate their dependability to the Federal Aviation Authority (FAA). If the probability of a system failure is higher than acceptable limits and can affect the safety of the passengers, the FAA will simply not approve of the system. In order to demonstrate that the system is dependable, many tests need to be run and the number of failures need to be documented. This is a costly and time consuming process.

**Question 4: Giving reasons for your answers, suggest which dependability attributes are likely to be most critical for the following?**

1. **An internet server provided by an ISP with thousands of customers**
2. **A computer controlled scalpel used in key hole surgery**
3. **A directional control system used in a satellite launch vehicle**
4. **An internet besed personal finance management system**

Dependability is an overarching concept for the following attributes: availability, reliability, safety, and security (Sommerville, 2011). Each of the four attributes can affect the dependability of the four examples above, but one attribute is most critical for each. Below I pair each example with its critical dependability attribute and my rationale for the choice of attribute.

I think that availability is the most important dependability attribute for an internet server provided by an ISP with thousands of customers. The service delivery up-time of the system is crucial here. If the customers experiences a lot of down time, they will simply change ISPs.

The most important dependability attribute for a computer controlled scalpel used in key-hole surgery is safety. Unless the probability of a system failure is low, no one will want to use the device and potentially risk human life or injury.

A directional control system used in a satellite launch vehicles needs to be reliable. If the direction control system is not delivering the functionality that the system expects, then the entire system will not work.

I think that security is crucial for an internet based personal finance management system because client confidentiality is paramount. If clients believe that their personal financial data will be hacked, they take their business elsewhere. The prevention of system intrusions is the critical dependability attribute here.

**References**

Brooks, F. P., Jr. (1995), “The Mythical Man Month.” Addison Wesley Longman, Inc. Retrieved from <http://my.uopeople.edu/mod/resource/view.php?id=57315>

Sommerville, I. (2011). Software Engineering. Edition 9. Retrieved from <http://my.uopeople.edu/mod/url/view.php?id=57311>