Engineering Ethics Cases with Numerical Problems

from an NSF & Bovay Fund sponsored workshop

August 14-18, 1995

Texas A&M University

Electrical Engineering Case 10

Software Testing

Authors: Suggested Courses:

Joseph Wujek All Design

Level:

Junior and Senior

I. Narrative

ìLetís test the software to be SURE it works!î

You are an engineer employed by Wondrous Avionics, Inc. (WA, Inc.). You are working on the project team developing the Mark-5, a new device in the prototype stage. The Mark-5 is a system for airplane attitude control combined with navigation. It has 116 input variables: X_1, X_2, ... X_{116}. Each X_i can take on any of the values permitted by a 32-bit word. Each X_i is sampled simultaneously for 20 nanoseconds every 1.30 milliseconds. Sampling at 1.30 milliseconds is the fastest possible due to the actuation and settling time of the electro-hydraulic mechanisms controlled by the Mark-5. Each of the possible totality of states of the X_i corresponds to one, and only one, configuration of aircraft control surfaces and resultant aircraft attitude, Y_j . Thus, $Y_j = f[X_1, X_2, ... X_{116}]$ in one-to-one correspondence. The Mark-5 outputs one value of Y_j in each 1.30 ms interval.

The output variable Y_j is generated by software, using a program which resides in firmware in the Mark-5. The software result actuates appropriate hardware drivers to actuate the hydraulic mechanisms.

In response to concerns from potential users of the Mark-5 regarding the use of software in safety-critical systems, the CEO of WA raises an issue at a project meeting. The CEO tells the project team, ìThe Mark-5 must be tested in all possible states to be *sure* that the software always works! Our customers are ënervousí about using software this way. I want us to answer their concerns by demonstrating, by test, that the Mark-5 gives the right output for each combination of inputs. After all, under some conditions the wrong output could cause a plane to crash!î

II. Numerical Problems

Problem 1. Assuming that the test speed is limited only by the 1.30 millisecond cycle time of the Mark-5, how long in hours would it take to perform the test desired by the CEO? Assume the test proceeds as fast as possible and without interruption, 24 hours/day, 7 days/week.

Problem 2. Same as (1), but assume three bugs were found, and each took two days to find and fix. Assume that the bugs were found at the 1/3, 2/3 and 99% complete points. The test must be run in its entirety after each bug fix.

Problem 3. WA estimates that it will cost $700/hour to run the test. Compute the cost of the test in part (a) and part(b).

Problem 4. Suppose it is decided that an 8-bit word is sufficient, instead of a 32-bit word. Also, make the robust(!) assumption that since itís only the software being tested, not the integrated system(!), the test cycle time can be reduced from 1.30 millisecond to 130 nanosecond. Rework part (1) with these design/test changes.

Problem 5. In your view, what would be a ìreasonableî test to run on the Mark-5 *system?*

III. Solutions to the Numerical Problems

1. This problem is intended to show the folly of attempting a deterministic approach to software testing. A ìbrute forceî analysis of all permutations of states yields an absurdly long test time. Therefore, a probabilistic approach must be employed, coupled with careful estimates of state-occupancies determined by detailed analyses of the input/output states and the software coding. (Doing so is beyond the scope here, but is extremely important in software engineering.) Even with these modern methods, test times are exceedingly long; and the process is expensive and complicated. In the end, one must accept a bounded risk, itself a risk!

A 32 bit word can take on 2^{32} = 4.295E9 states (rounding to four SF). Each of the 116 variables may take on any of these values. So, the number of distinct states is: S = $(4.295E9)^{116}$. Logarithms may be used to find: S = 2.653(E1117) possible states.

The interval between sampling is 1.30 ms, so the test-time is:

T = (2.653)(E1117)(1.30E-3) = 3.449(E1114) seconds, or **9.6(E1110) hours**.

Based on 24 hours/day, 365 days/year testing it would thus require a *minimum* (no restart from zero after bug fixes) of **1.1(E1107) years to perform the 100% test!** For perspective(?), the age of the universe is estimated to be of the order of E10 years.

2. The 2 days/bug to fix bugs is negligible compared to the test time, to say the least! Thus the accumulated test time is: T = [9.6(E1110)hours][(1/3) + (2/3) + (0.99) + (1)] = **2.9(E1111) hours**.

3. 2.9(E1111 h)($700/h) = $2.0E1114.

4. An 8-bit word has 2^{8} = 256 states. Then $(256)^{116}$ = 2.27E279 possible states exist. Test time is:

T = (1.30E-7)(2.27E279) = 2.95E272 seconds, or **8.2E268 hours,** still an impossible test!

5. From what is given in the problem statement, no ìreasonableî test exists. If it is technically possible to test each

word in parallel, then somehow combine results in some manageable form, a test may be possible. But such artifices are dependent upon engineering judgment and may not yield a thorough test and reliable test..

IV. Ethics Problem

Should the Mark-5 be built and offered for sale without any software testing? What ethical principles are involved in your evaluation?

V. Solutions to Ethical Problems

Based on the results of the above analyses, particularly part 5, the Mark-5 should not be sold without some software testing. To deploy the Mark-5 without some form of software testing, when failure endangers human lives, violates several moral principles.

First, it violates the principle of informed consent, which says that people should be allowed to give their informed consent to dangers, especially when there is a danger of death. Many of the users of the aircraft would probably be unaware of the problems of the Mark-5 or incapable of evaluating the technical issues and so not understand the seriousness of the problem. Thus, they would not be giving informed consent to an unusual danger. Even if they were aware of the danger, they would probably have no choice but to use the Mark-5, if was installed on their aircraft. Thus, they would not be giving consent to the unusual danger.

Second, selling the Mark-5 without testing violates the Golden Rule, which requires that if others perform an action like our own, we must be willing to accept the consequences of the action. The manufacturers would probably not want to fly in an aircraft equipped with the untested Mark-5, knowing what they know about its problems. Given this, they should not impose this danger on others.

Third, deploying the untested Mark-5 would violate what some have called the "New York Times Test." Ask yourself whether you, as the CEO of Wondrous Avionics, would be willing to have it generally known that your company sold the Mark-5, knowing the answers to questions 1-5. Your answer would probably be that you would not. Therefore, you have no right to impose this risk on others.

Fourth, selling the untested Mark-5 would probably violate the test of Rule Utilitarianism. Would it maximize utility or general well-being if every manufacturer sold items with as much potential for disaster as the Mark-5? The answer is almost certainly that it would not. If manufacturers did this as a general rule, many accidents would result and the confidence of air flight and probably in technology generally would be eroded. This would not lead to general well-being or welfare, but in fact would be socially harmful.