Abstract

1. Context

Software based systems have several advantages over hardware based systems in terms of functionality, cost, flexibility, maintainability, reusability, etc. However, software is prone to failure. Poorly written safety-critical software may lead to catastrophic failures and life threatening situations. Hence, safety-critical software must be adequately tested; and the probability of occurrence of software failures must be studied.

Quantification of software reliability is considered an unresolved issue; and existing approaches and models have assumptions and limitations which are not acceptable for safety applications. Also, to build reliable software, it is necessary to study the factors which are likely to affect the software reliability.

2. Objectives

1. To propose an automated method to generate test cases, and to determine test adequacy in safety-critical software.

2. To propose an approach to quantify software reliability in safety-critical systems of nuclear reactors.

3. To study the factors affecting software reliability in such safety systems.

4. To understand the relationship between the software reliability and number of faults remaining in the software.

5. To understand the relationship between the software reliability and safety in safety critical systems.
3. Method

To quantify the software reliability, a hybrid approach using software verification and mutation testing is proposed. Techniques to solve related issues such as quantification of software test adequacy and detection of equivalent mutants are also presented. The steps proposed to quantify software reliability are:

1. Generation of large number of test cases, where each test case has a unique execution path. To achieve this, code coverage information and genetic algorithms are used.

2. Verification of test cases using a semi-formal model, which is traceable to requirements; and acts as a test oracle.

3. Calculation of test adequacy for the above generated test cases in the range [0,1] using mutation score and conservative test coverage.

4. Calculation of software reliability using the computed test adequacy and the amount of verification carried out.

The formulae for software reliability are derived, and the factors affecting software reliability are presented. The proposed methods are applied to software in the following instrumentation and control systems for fast breeder reactors:

1. Fresh Sub-assembly Handling System (FSHS)

2. Reactor Startup system (RSU)

3. Steam Generator Tube Leak Detection system (SGTLD)

4. Core Temperature Monitoring System (CTMS)

5. Radioactive Gaseous Effluent System (GES)

6. Safety Grade Decay Heat Removal system (SGDHR)

Also, for each case study, mutant characteristics during mutation testing, and the relationship between software reliability and safety are presented.
4. Major results

1. For the case studies, the proposed test case generation technique has resulted in high test adequacy. Using the generated test cases, the probability of software failure in the case studies has been demonstrated to be $< 10^{-5}$ for a random input from the input domain, with 95% confidence level.

2. In mutation testing, for an effective set of test cases, the unkillled mutants have been found to have lower variance in their properties when compared to the killed mutants.

3. Three factors: (i) test adequacy, (ii) the amount of verification carried out, and (iii) the amount of verified code reused; have been found to be affecting the software reliability.

4. The results of present study suggest that software reliability estimates based on the number of faults present in the software alone, are likely to be inaccurate for safety-critical software.

5. The empirical results indicate that: for safety-critical software, the required safety can be achieved by improving the reliability; however the vice-versa is not always true.

5. Conclusion

The methods and analysis presented in this thesis demonstrate the use of software testing to arrive at an estimate of the software reliability. The results on relationship between the software reliability and safety in safety-critical systems would be helpful in understanding the dynamics behind developing safer software based systems.

The proposed approaches can be used by safety-critical software developers to improve the software reliability. Also, the regulators may use the techniques to verify reliability, safety, and dependability claims.