Calibrating Software Reliability Models

9.1 Introduction

Estimating software reliability measures that will be perceived by users is important in order to decide when to release software. Usually, software reliability models are applied to system test data with the hope of estimating the failure rate of the software in user environments. This chapter discusses recent methods and research on how to quantify the mismatch between the system test environment and the field environment based on recent studies (Zhang 2002; Teng 2001). The chapter also discusses a generalized random field environment (RFE) model incorporating both testing phase and operating phase in the software development cycle for estimating the reliability of software systems in the field. Examples are included to illustrate the calibrating software reliability model based on test data.

**Notation**

- $a(t)$: Fault content function, \textit{i.e.}, total number of faults in the software including the initial and introduced faults
- $b(t)$: Fault detection rate function (faults per unit of time)
- $b_{\text{test}}$: Average per fault failure rate during system test interval
- $b_{\text{field}}$: Average per fault failure rate in the field
- $\bar{b}_{\text{test}}$: Long-term average per fault failure rate during system test interval
- $\bar{b}_{\text{field}}$: Long term average per fault failure rate in the field
- $\lambda(t)$: Failure intensity function (faults per unit of time)
- $\tilde{\lambda}(T)$: Failure intensity representation based on system test data
- $K$: Calibration factor
- $m(t)$: Mean value function, \textit{i.e.}, the expected number of faults detected by time $t$
- $N(t)$: Number of detected faults by time $t$
- $\bar{N}(t)$: Number of residual faults by time $t$
- $T$: Duration of system test interval
9.2 Calibration Factor Approach

Let us assume that the system test ends at time \( T \) and after that the software is delivered to the field. The expected number of faults detected and removed by time \( T \) is \( m(T) \). To account for the mismatch between the system test field environments, Zhang et al. (2002) recently proposed linking the error detection rate function \( b(t) \) under the system test environment, say \( b_{\text{test}}(t) \), to a different \( b(t) \) under the field environment, say \( b_{\text{field}}(t) \). Define

\[
\overline{b}_{\text{test}} = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} b_{\text{test}}(t) \, dt
\]  

(9.1)

Intuitively, \( \overline{b}_{\text{test}} \) represents the long-term average per fault failure rate during system test. Using an analogous definition for \( \overline{b}_{\text{field}} \), Zhang et al. (2002) defined the calibration factor as the ratio \( K = \overline{b}_{\text{test}} / \overline{b}_{\text{field}} \). In the case where the system test and field environments are the same, \( K \) will be unity.

Assuming that the fault detection rate, \( b_{\text{test}}(t) \), for system test environments is given by

\[
b_{\text{test}}(t) = \frac{b_{\text{test}}}{1 + \beta e^{-\beta t}}
\]  

(9.2)

where \( \beta \) represents a learning parameter and \( b_{\text{test}} \) is the limiting value of the fault detection rate. Note that \( \beta = 0 \) coincides with no learning, and the fault detection rate reduces to the constant value \( b_{\text{test}} \).

A General Approach (Zhang 2002)

Consider a context where only system test data is available for a release and it is desired to estimate the field failure rate of the software. Assume that system test and field data of the previous releases of the same product or similar product are also available from which a \( K \) factor can be obtained. We suppose an NHPP SRGM model (see Chapter 6) has been fit to the system test data and the assumptions underlying GO model (i.e., no learning factor and no introduction of new faults) are adequately satisfied in the field environment. The software failure rate in the field can be estimated by the following steps:

1. Estimate the calibration factor \( K \) from previous releases/projects.
2. Estimate the number of residual faults based on system test data, \( \hat{N}(T) = \hat{a}(T) - N(T) \), and the long-term average per fault failure rate \( \overline{b}_{\text{test}} = \hat{b}_{\text{test}} \) from the system test data.
3. Calibrate the system test analysis to estimate the average per fault failure rate in the field using the calibration factor \( K \). The average per fault failure rate in the field is estimated by \( \hat{b}_{\text{field}} = \hat{b}_{\text{test}} / K \).