Software Reliability II

1

Content

- How to predict the number of faults?
- How to record failures?

2

Prediction of the Number of Faults

Various prediction methods

- error seeding
- independent testing

3

Error Seeding Method

- N denotes the total number of faults (N is fixed but unknown)
- Suppose M faults are seeded into the program

After testing

- n denotes the number of original faults detected
- m denotes the number of seeded faults detected

4

Error Seeding Method (continued)

- Assumption – original and seeded faults are of the same chance of being detected
- Therefore, we should have $(n/N) = (m/M)$

5

Error Seeding Method (continued)

Example:

- Seed in 50 distinct faults
- After testing, identify 15 faults
- Amongst the 15 identified faults, 10 are seeded faults
- How many faults are still in the program

6

Error Seeding Method (continued)

Example:

- n = 5
- m = 10
- M = 50

(n / N) = (m / M)

(5 / N) = (10 / 50)

N = 25

7

Error Seeding Method (continued)

Example:

Number of faults still remain in the program

= 25 - 5

= 20

8

Error Seeding Method (continued)

Problem:

- How to seed the appropriate faults?

9

Independent Testing Method

- Two independent testing teams
- N_1 and N_2 denote the number of faults detected by these two teams
- N_{12} denotes the number of faults that were detected by both teams
- N denotes the total number of faults prior to the testing (N is fixed but unknown)

10

Independent Testing Method (continued)

Assumption – every fault has the same chance of being detected

Effectiveness of Team 1 = (N_1 / N)

Also, effectiveness of Team 1 = (N_{12} / N_2)

11

Independent Testing Method (continued)

Therefore, we have $(N_1 / N) = (N_{12} / N_2)$

That is, $N = (N_1 N_2) / N_{12}$

12

Independent Testing Method (continued)

Example

- Team 1 identified 12 faults
- Team 2 identified 18 faults
- 6 faults are commonly identified

13

Independent Testing Method (continued)

Example

$N = (N_1 N_2) / N_{12}$

$= (12 * 18) / 6$

$= 36$

Number of remaining faults

$= 36 – (12 + 18) + 6 = 12$

14

Definition of Software Reliability

Probability for a software
- without failure
- a period of operation time
- a specified operation environment

Example: A system has a reliability of 0.96 for 12 hours when used by the average user

15

Classifications of Failure Severity

Some classifications on failure severity

(1) Minor – not causing injury
(2) Marginal – minor injury, mission degradation
(3) Critical – severe injury, mission loss
(4) Catastrophic – death

16

Reliability versus Testing

Random Testing

- Uniform distribution/profile
- Operational distribution/profile

17

Example

Consider a program with 20 inputs.

- uniform distribution – each input has 0.05 probability of being selected as a test case
- operational distribution – each input has a probability of being selected as a test case, which is equal to its probability of being used in operation.

18

Basic Definitions

- MTTF – mean time to failure
- MTTR – mean time to repair
- MTBF – mean time between failures

MTBF = MTTF + MTTR

19

Failure Occurrences

- Time of failure
- Time interval between failures
- Cumulative failures experienced up to a given time
- Failures experienced in a time interval

20

Example

- Suppose failures occur at the following times (in seconds):

  11, 17, 22, 39, 51, 66, 78, 93, 123, 145, 167, 196, 234, 265, 293, 303, 335, 389, …

21

Time-Based Failure Specification

11, 17, 22, 39, 51, 66, 78, 93, 123, 145, 167, 196, 234, 265, 293, 303, 335, 389, …

Failure Number	1	2	3	4	5	6	7	8	9
Failure Time (in sec)	11	17	22	39	51	66	78	93	123
Failure Interval (in sec)	11	6	5	17	12	15	12	15	30

22

Time-Based Failure Specification

11, 17, 22, 39, 51, 66, 78, 93, 123, 145,
167, 196, 234, 265, 293, 303, 335, 389, …

Failure Number	10	11	12	13	14	15	16	17	18
Failure Time (in sec)	145	167	196	234	265	293	303	335	389
Failure Interval (in sec)	22	22	29	38	31	28	10	32	54

23

Time-Based Failure Specification

Failure Number	1	2	3	4	5	6	7	8	9
Failure Time (in sec)	11	17	22	39	51	66	78	93	123
Failure Interval (in sec)	11	6	5	17	12	15	12	15	30

Failure Number	10	11	12	13	14	15	16	17	18
Failure Time (in sec)	145	167	196	234	265	293	303	335	389
Failure Interval (in sec)	22	22	29	38	31	28	10	32	54

24

Failure-Based Failure Specification

11, 17, 22, 39, 51, 66, 78, 93, 123, 145,
167, 196, 234, 265, 293, 303, 335, 389, …

Suppose we choose a period of 30 seconds

Time (in sec)	30	60	90	120	150	180	210	240	270	300	330	360	390
Number of Failures in interval	3	2	2	1	2	1	1	1	1	1	1	1	1
Cumulative Failures	3	5	7	8	10	11	12	13	14	15	16	17	18

25

Failure-Based Failure Specification

11, 17, 22, 39, 51, 66, 78, 93, 123, 145,
167, 196, 234, 265, 293, 303, 335, 389, …

Time (in sec)	30	60	90	120	150	180	210	240	270	300	330	360	390
Number of Failures in interval	3	2	2	1	2	1	1	1	1	1	1	1	1
Cumulative Failures	3	5	7	8	10	11	12	13	14	15	16	17	18

26

Failure-Based Failure Specification

11, 17, 22, 39, 51, 66, 78, 93, 123, 145,
167, 196, 234, 265, 293, 303, 335, 389, …

Time (in sec)	30	60	90	120	150	180	210	240	270	300	330	360	390
Number of Failures in interval	3	2	2	1	2	1	1	1	1	1	1	1	1
Cumulative Failures	3	5	7	8	10	11	12	13	14	15	16	17	18

27

Two Measures

- mean value function – average cumulative failures
- failure intensity function – number of failures per unit time

28

Probability Distribution of Failures within a Time Period of Execution Time

Number of Failures	0	1	2	3	4	5	6
Probability	0.05	0.12	0.25	0.28	0.15	0.11	0.04

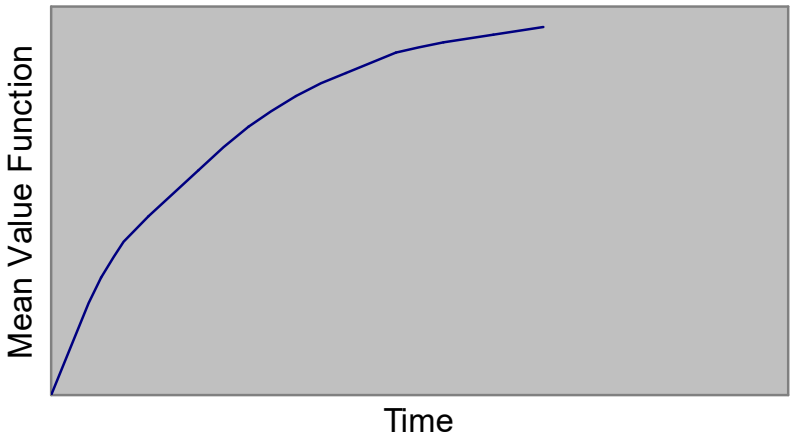
Mean is 2.85 failures

29

Example

Time	1	2	3	4	5	6	7	8	9	10
Mean Value Function	3.0	4.6	5.8	6.9	7.7	8.3	8.8	9.1	9.3	9.5

30



31

Summary

32