Agenda

- Software Reliability Engineering
  - Chapter 6: Guiding Test

This lecture provides reference material for the book entitled "Software Reliability Engineering: More Reliable Software Faster and Cheaper" by John D. Musa © 2004

This lecture material is copyrighted by Laurie Williams (2007). However, you are encouraged to download, forward, copy, print, or distribute it, provided you do so in its entirety (including this notice) and do not sell or otherwise exploit it for commercial purposes.

For PowerPoint version of the slides, contact Laurie Williams at williams@csc.ncsu.edu.

© 2007 Laurie Williams

When to release the product

1. Terminate test satisfactorily for the base product with the failure intensity to failure intensity objective (FI/FIO) ratio at 0.5 or less
2. Terminate test satisfactorily for all the product variations, with their FI/FIO ratios not appreciably exceeding 0.5
3. Accepted the product and its variations in any acceptance test rehearsals planned for them.
4. Accepted all supersystems.
5. Resolved all outstanding (usually Sev 1 and 2) failures.

© 2007 Laurie Williams

Estimating failure intensity

- Make periodic estimates of FI/FIO based on failure data using a software reliability estimation program, such as CASRE.
- Software reliability estimation programs are based on software reliability estimation models (such as the Musa-Basic model) and statistical inference.
- Guideline:
  - Estimate weekly if more than 3 months until release
  - Estimate semi-weekly if 1-3 months until release
  - Estimate daily if less than one month until release

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

Musa models

- Musa Basic
  - Assumes finite failures in infinite time
  - Tends to be optimistic (low) in estimating FI/FIO
  - Use if: product is stable (not changing or evolving as test proceeds), has a very low FIO, and a long test period [so you can expect that all but a very small number of failures can be removed by the end of system test]
- Musa-Okumoto logarithmic
  - Assumes infinite failures
  - Tends to be pessimistic (high) in estimating FI/FIO
  - Use if: characteristics not as specified above

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

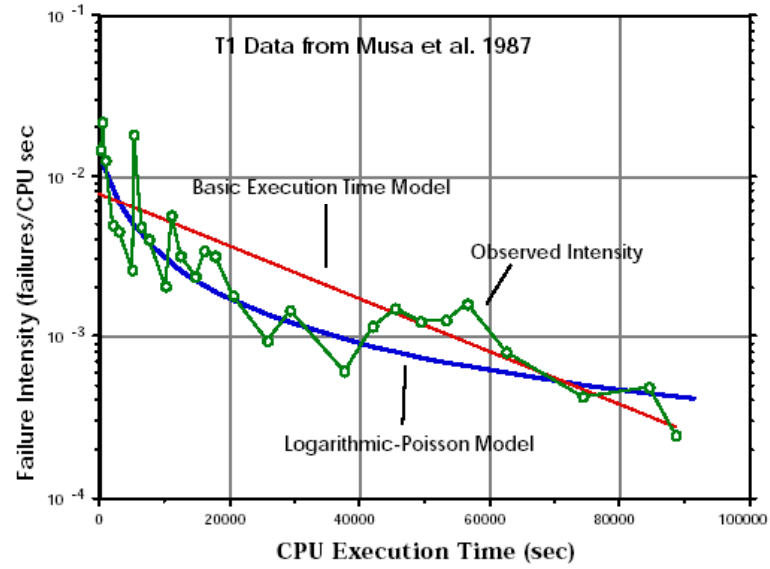

Figure 1. Empirical and modeled failure intensity.

© 2007 Laurie Williams

Figure from Vouk, M. Software Reliability Engineering, 2000 Annual Reliability and Maintainability Symposium

Selecting software reliability estimation model

- Model simplicity
  - Should be understandable by software engineer without extensive mathematical background
- Model maturity
  - Well developed model that has been applied broadly with real data and given reasonable results

© 2007 Laurie Williams

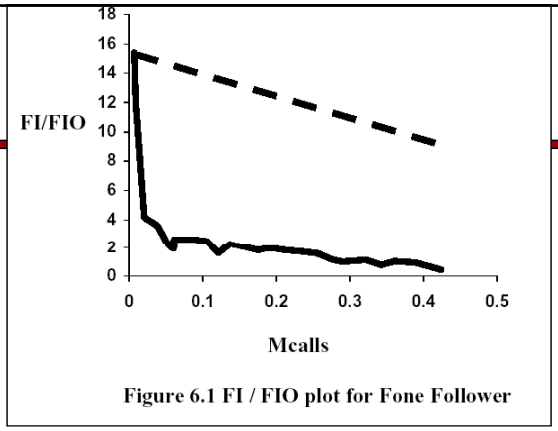

Figure 6.1 FI / FIO plot for Fone Follower

Sharp drop is typical when you drive test with the operational profile – invoke most frequently-used operations first and remove these faults efficiently.

Dotted line more typical of system test when all operations are tested with equal probability.

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

Figure from Musa, J., *Software Reliability Engineering*, 2004.

Estimation program: CASRE
(Computer-Aided Software Reliability Estimation)

- Estimation based upon underlying reliability models
- Good results when estimating with programs with 5000 or more SLOC; satisfactory results when estimating with programs of 1000 or more SLOC.

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

For each execution of CASRE

- When the FI/FIO ratio is large and the trend indicates little chance of achieving FIO by the schedule date, consider one or more of the following actions:
  - Defer features
  - Adjust the balance of failure intensity, delivery date, and cost objectives
  - Add additional test and debugging resources by employing overtime
- If FI/FIO > 2 and shows significant upward movement over several data points, determine the cause
  - Unexpected system evolution due to new developed code
    - Change control!
  - Varying test selection probabilities
    - Operations tested late may have unpleasant surprises that threaten schedule
- Continue software reliability growth test until FI/FIO drops to 0.5 or less.

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

Certifying Reliability

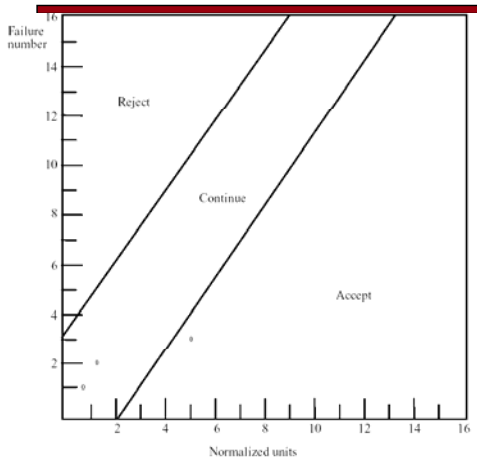

Figure 6.2 Reliability demonstration chart

For a given:

•Discrimination ratio or factor of error in estimating FI you are willing to accept (2 in figure)

•Consumer risk or the probability you are willing to accept of falsely saying the FIO is met when it is not (0.1 in figure)

•Supplier risk or the probability you are willing to accept of falsely saying the FIO is not when it is (0.1 in figure)

•Normalize failures by multiplying by FIO in same units
ex: failure 1 occurred on call 187,500; normalized =
.1875 Mcalls* 4 failures/Mcalls = 0.75

Research Group Software Engineering @ NCSU © 2007 Laurie Williams NC STATE UNIVERSITY

Figure from Musa, J., *Software Reliability Engineering*, 2004.

To certify or not to certify?

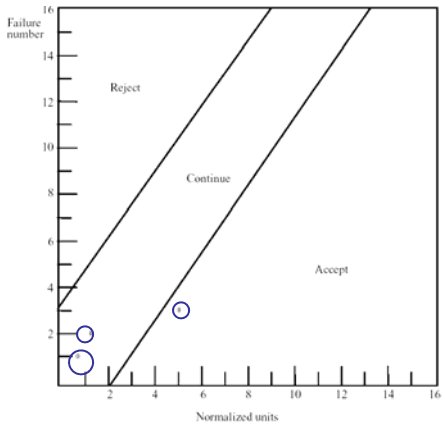

Figure 6.2 Reliability demonstration chart

Table 6.1 Failures in certification test of operating system for Fone Follower

Failure number	Measure (M calls)	Normalized units
1	0.1875	0.75
2	0.3125	1.25
3	1.25	5

You can certify or reject as soon as you cross the accept or reject boundary

Schedule driven: increase the risk values and take more risk

Reliability/availability driven; decrease the risk values to reduce the risk

© 2007 Laurie Williams

Figure and Table from Musa, J., *Software Reliability Engineering*, 2004.

If you're still in Continue region at end of test . . .

- Note the number of defects that have been found.
- Go horizontally across from the number of defects that have been found on the y axis to the "continue/accept" boundary line. Find the x position for that intersection.
- Divide this x position by the normalized units for the last defect. This provides you a multiplier. You multiply the FIO by this multiplier to get the achieved FIO. You decide if you want to keep that or to extend the testing duration by this multiplier.

© 2007 Laurie Williams

Summary

- You use system test failure data to drive your decision making:
  - Ready to release developed software? See if it is estimated that you have a FI/FIO ratio of 0.5 or less.
  - Able to certify/accept purchased software? See if your failure data put you in the “Accept” region of a reliability demonstration chart.

RESEARCH Group Software Engineering @ NCSU
© 2007 Laurie Williams
NC STATE UNIVERSITY