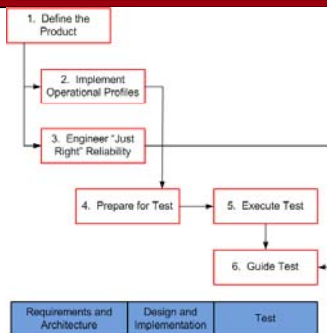


## SRE: Problem, Process, Product



This lecture provides reference material for Chapter 1 of 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.

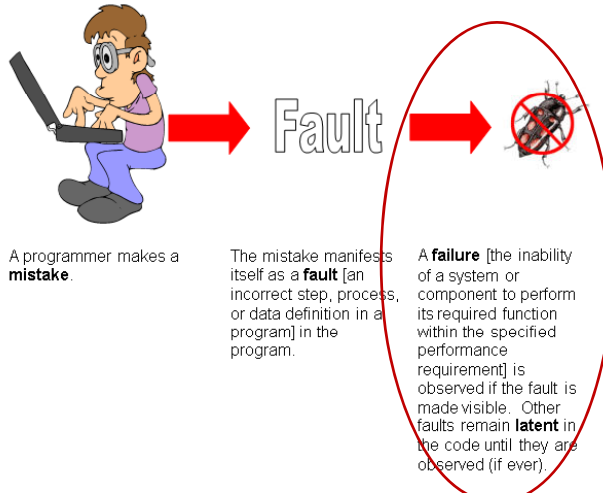
For PowerPoint version of the slides, contact Laurie Williams at [williams@csc.ncsu.edu](mailto:williams@csc.ncsu.edu).

## Software Reliability Engineering (SRE)

- "Engineer" reliability
  - Develop a product in such as way that the product reaches the "market" at the right time, at an acceptable cost, and with satisfactory reliability [work on the balance between the three]
  - User-oriented rather than developer-oriented
- Objectives:
  - Help engineers, managers or users or software make more precise decisions
  - Make everyone more concretely aware of software reliability by focusing attention on it



## Faults vs. Failures



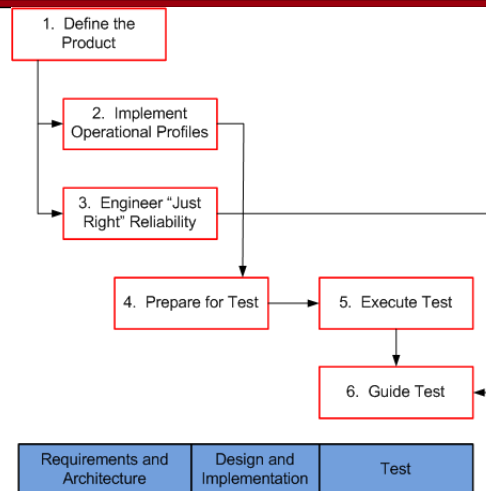
## Reliability

- **Reliability**: the probability that a system will continue to function without failure for a specified number of natural units or a specified time
  - Correctness, safety, operational aspects of usability and user-friendliness
  - “time” may be in natural or time units
    - Examples of “natural” units – runs, pages of output, transactions, telephone calls, jobs, semiconductor wafers, queries, API calls
  - Failure intensity = failures per natural or time unit

## Availability

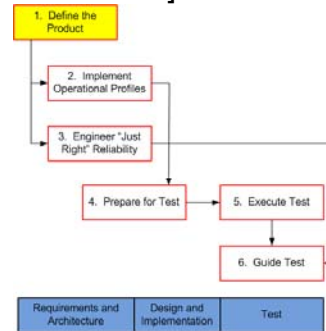
- **Availability:** average (over time) probability that a system is currently functional in a specified environment OR ratio of uptime to the sum of uptime plus downtime
  - Downtime for a given interval is the product of the length of the interval, the failure intensity, and the mean time to repair (MTTR)
    - $10 \text{ hours} * .1 \text{ failures/hour} * .5 \text{ hours/failure} = .5 \text{ hours}$
  - MTTR is average time required to restore the data for a program, reload the program, and resume execution
  - $\text{Availability} = 9.5/10 = .95 = 95\%$

## SRE Process



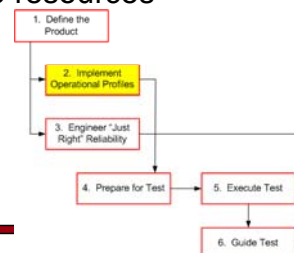
## 1. Define the product

- Establish who the supplier, customers, and users are
- List the associated systems [user can't separate out new system from whole system when a problem occurs]



## 2. Implement Operational Profiles

- An **operational profile** is a complete set of the operations (major system logical tasks) of a system with their probability of occurrence [e.g. the requirements and the probability of how often the users will use each requirement/scenario].
- You can use the operational profile to prioritize all aspects of development and to allocate resources accordingly.



## Operational Profile Example

- Suppose you have a system with operations A and B. Operation A executes 90% of the time and Operation B, 10%. Assume each operation has 10 faults and you have 10 hours of test and debugging effort available. Finding and fixing each faults requires one hour of effort.
  - How many “operational” faults if you spend 5 hours on each?
    - Spend equal amount of time on each
    - 5 faults remain in each
    - $.9(5) + .1(5) = 5$  “operational” faults (faults likely to be encountered by customer)
  - How many “operational” faults if you spend your time relative to the operational profile?
    - Spend 9 hours on A, 1 hour on B
    - 1 fault remains in A, 9 faults remain in B
    - $.9(1) + .1(9) = 1.8$  “operational” faults

## 3. Engineer “Just Right” Reliability

- Define failure in project-specific terms with examples
- Choose a common reference unit for all failure intensities (for example, failures with respect to hours, calls, or pages)
- Setting a system failure intensity objective (FIO) for each associated system
- Engineer strategies to meet the FIO

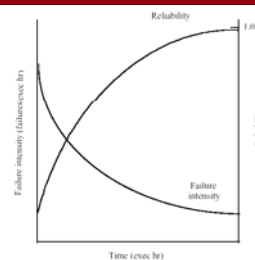
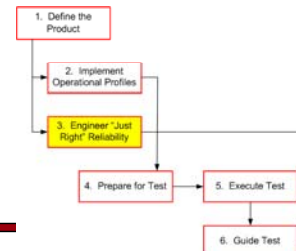
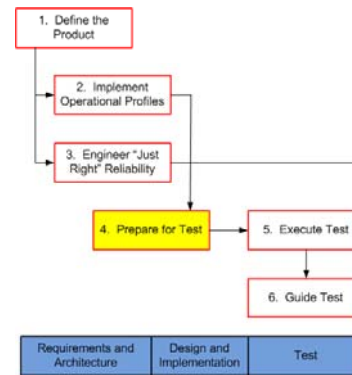


Figure 1.2 Reliability and failure intensity



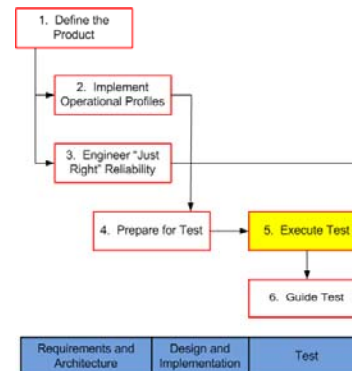
## 4. Prepare for Test

- Specify black box test cases for the new operations of the system and variations according to operational profile
- Allocate test time



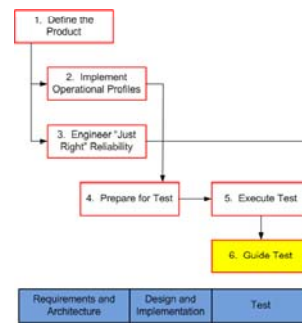
## 5. Execute Test

- Invoke test
- Identify system failures



## 6. Guiding Test

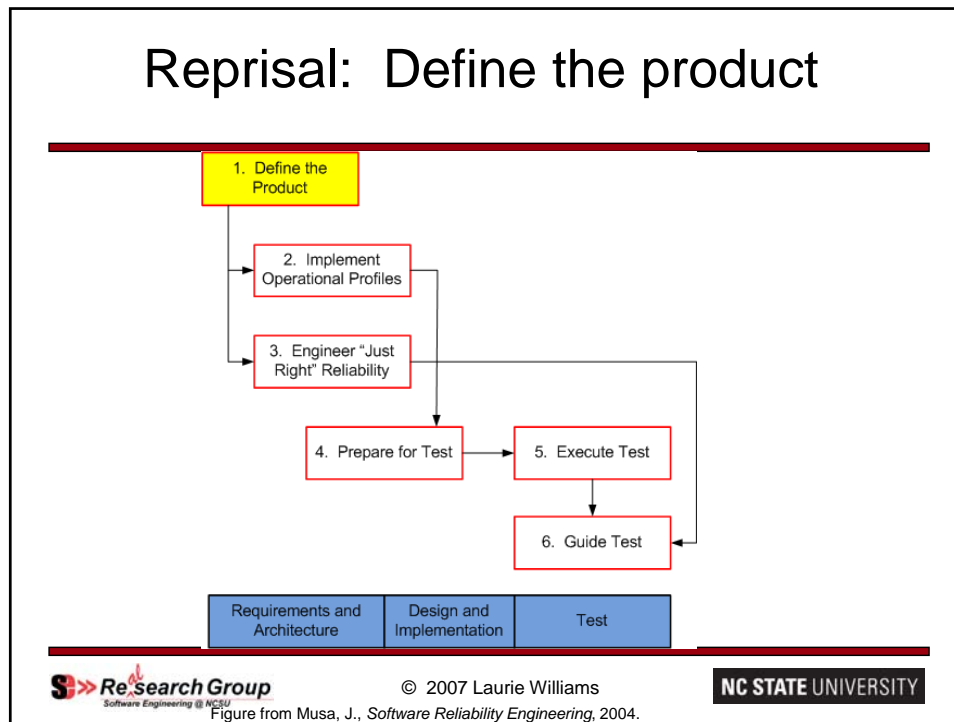
- Processing failure data gathered in test
- Track reliability growth (where reliability growth is defined as the improvement in reliability that results from the correction of faults) or reduction in failure intensity
- Certify the reliability of the base product and its variations that customers will acceptance test. Certification is a go/no-go decision.



## During Post Delivery and Maintenance

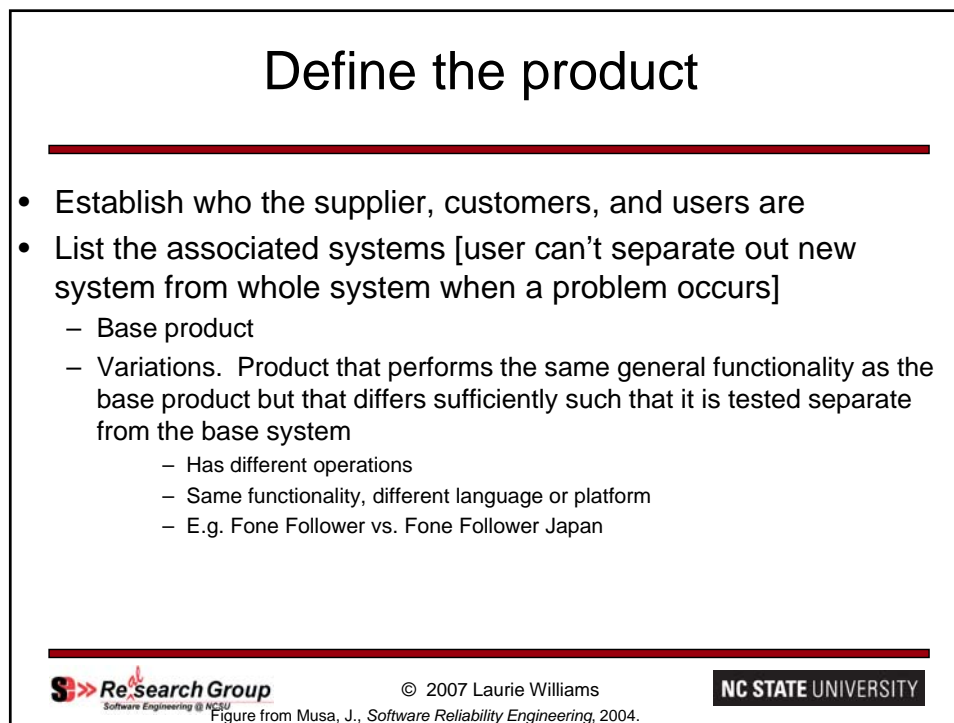
- Determine the actual reliability achieved
- Determine the actual operational profile experienced
  - Affects the engineering of “just right” reliability for the next release.
  - Build recording and reporting mechanisms into the product.
- Collect data on failure intensity and customer satisfaction.

## Reprisal: Define the product



## Define the product

- Establish who the supplier, customers, and users are
- List the associated systems [user can't separate out new system from whole system when a problem occurs]
  - Base product
  - Variations. Product that performs the same general functionality as the base product but that differs sufficiently such that it is tested separate from the base system
    - Has different operations
    - Same functionality, different language or platform
    - E.g. Fone Follower vs. Fone Follower Japan



## Determination of Supersystems to Test

Table 1.5 Supersystem table

Operating system	Server	Network
Windows XP	S1	N1
Windows XP	S1	N2
Windows XP	S2	N1
Windows XP	S2	N2
Macintosh	S1	N1
Macintosh	S1	N2
Macintosh	S2	N1
Macintosh	S2	N2

Enumerate all possible combinations of printer that must work with Windows XP and Mac with servers S1 and S2 and networks N1 and N2.

Operational Profile  
Windows: 0.9

Mac: 0.1

Table 1.6 Use of supersystems

Oper. sys.	Server	Network	Probability
Windows XP	S1	N1	0.504
Windows XP	S1	N2	0.216
Windows XP	S2	N1	0.126
Macintosh	S1	N1	0.056
Windows XP	S2	N2	0.054
Macintosh	S1	N2	0.029
Macintosh	S2	N1	0.014
Macintosh	S2	N2	0.006

S1: 0.8

S2: 0.2

N1: 0.7

N2: 0.3

## Background: Characterize failure occurrences

- Time of failure
- Time interval between failures
- Cumulative failures experienced up to a given time
- Failures experienced in a time interval
- **Random** (unpredictable) variables → we do not know the values with certainty; so many factors affect the values that it is impractical to predict them – we don't know when the next failure will occur (depends on where the faults are and what the customer does)

## Background: Characterize failure occurrence I

Table 1.1 Time-based failure specification

Failure	Failure time	Failure interval
1	10	10
2	19	9
3	32	13
4	43	11
5	58	15
6	70	12
7	88	18
8	103	15
9	125	22
10	150	25
11	169	19
12	199	30
13	231	32
14	256	25
15	296	40

Mean time to failure (MTTF)  
= average value of next failure interval

Mean time to repair (MTTR)  
= average amount of time to fix the product and put it back into operation

Mean time between failures (MTBF) = sum of MTTF and MTTR

## Background: Characterize failure occurrence II

- Cumulative failures experiences up to a given time
- Failures experienced in a time interval

Table 1.2 Failure-based failure specification

Time (sec)	Cumulative failures	Failures in interval
30	2	2
60	5	3
90	7	2
120	8	1
150	10	2
180	11	1
210	12	1
240	13	1
270	14	1

## Background: Probability Distribution of Failures

Table 1.3 Typical probability distribution of failures

Value of random variable (failures in time period)	Probability	Product of value and probability
0	0.10	0
1	0.18	0.18
2	0.22	0.44
3	0.16	0.48
4	0.11	0.44
5	0.08	0.40
6	0.05	0.30
7	0.04	0.28
8	0.03	0.24
9	0.02	0.18
10	0.01	0.1
Mean failures	---	3.04

## Summary

- The focus of SRE is on prioritizing development effort and testing such that the customer views the product as having higher reliability
- All faults are not equal
  - Some could feasibly remain latent for the life of the product
  - Others will likely be revealed in normal use
  - Focus on faults that are more likely to become failures