HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Software Quality Assurance
Đảm bảo chất lượng phần mềm

Lecture 8: Software Reliability

Contents

- Definition of Reliability
- Reliability Criteria
- Operation Profile

8.1. Reliability

Definition of Reliability (1)

- The **probability of failure-free operation of a software system for a specified time in a specified environment**
- The key elements:
  - Probability of failure free operation
  - Length of time of failure free operation
  - A given execution environment

Definition of Reliability (2)

- **Failure intensity** is a measure of the reliability of a software system operating in a **given environment**
- The lower the failure intensity of a software system, the higher is its reliability

What is the difference between two Definitions?

Failures vs Faults

- Software Failure: an incorrect result with to the specification or unexpected software behavior perceived by users
- Software Fault: identified or hypothesized cause of the software failure
- Failure – Effect vs Fault – Cause
- Defect: generic term referring to both failure and fault

Time Intervals

- Execution time
- Calendar time
- Clock time
- What is the most relevant ?

- The CPU time that is actually spent by the computer in executing the software
- The time people normally experience in terms of years, months, weeks, days…
- The elapsed time from start to end of computer execution in running the software

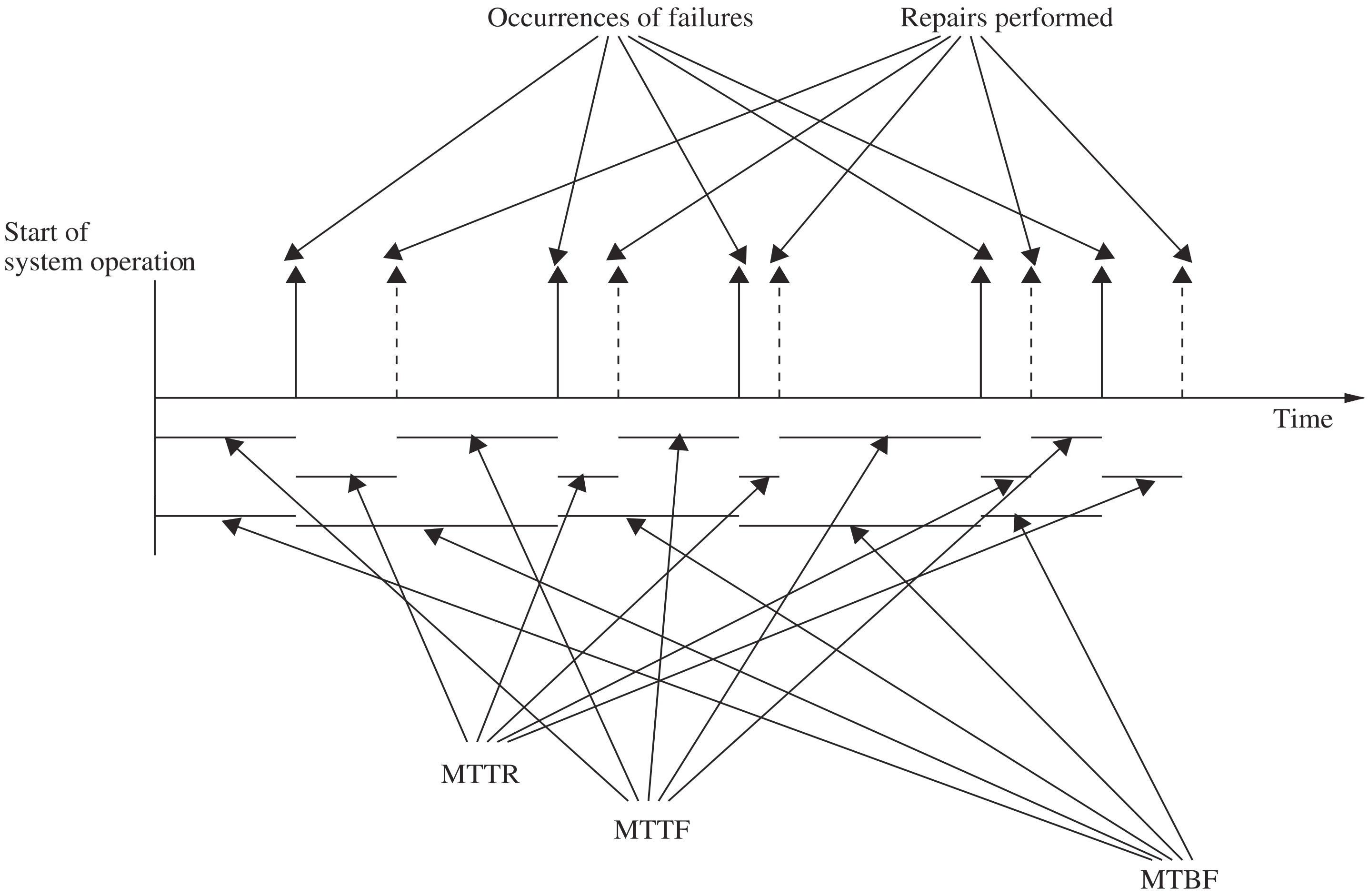
Questions to estimate the Software Reliability

- Questions about the Reliability in terms of time and failure:
    - What is the time interval between two successive failures?
    - How many failures have been observed within a certain time interval, for example in the past one month or one year?
    - What is the total number of failures observed so far?
- Answer these questions give an indication of the quality level of a software

Time Interval between Failures

- What does the small time interval between successive failures tell?
    - System is failing frequently and the reliability level is too low
- Reliability metrics
    - Mean Time to Failure (MTTF)
    - Mean Time to Repair (MTTR)
    - Mean Time between Failures (MTBF)
    - MTBF = MTTF + MTTR

Relationship between MTTR, MTTF, MTBF



8.2. Mathematical Model of Reliability

Mathematical Modelling of Software Reliability

- Motivation
  - Counting the number of failures observed so far and plotting the data as a function of time to express the change in the reliability of the system
  - A rising graph of the **cumulative number of failures** shows that there are more faults in the system
  - The rate of rising of the graph is the rate at which failures are being observed
  - What is the meaning of small rate of rising of graph ?

Mathematical Functions of Failures

- Cumulative failures are counted in periodic intervals
  - Count cumulative failures every month
- Time: CPU time (τ) or calendar time (t)
- Cumulative failure function μ (τ)
  - The total number of failures observed until execution time τ from the beginning of the system execution
- Failure intensity function λ (τ)
  - The number of failures observed per unit time after τ time units of executing the system from the beginning
- $\lambda(\tau) = \frac{d\mu(\tau)}{\mathrm{d}\tau}$ $\mu(\tau) = \int_0^\tau \lambda(x)\mathrm{d}x$

8.3. Factors

Factors influencing software reliability

- The software reliability depends on two categories of information:
  - The way users operate the system – also known as ***operational profile***
  - The number of faults present in the software system
    - ***Size and Complexity of code***
    - ***Characteristics of Development Process***
    - ***Education, Experience, and Training of Personnel***
    - ***Operational environment***

Applications of Software Reliability

- Comparison of Software Engineering Technologies
  - What is the cost of adopting technology?
  - How does the new technology affect the development schedule?
  - What is the return from the new technology in terms of software quality?
- Measuring the Progress of System Testing
  - Percentage of test cases executed
  - Percentage of successful execution of high-priority functional tests
  - Controlling the System in Operation
  - Better insight into Software Development Process

8.4. Operational Profile

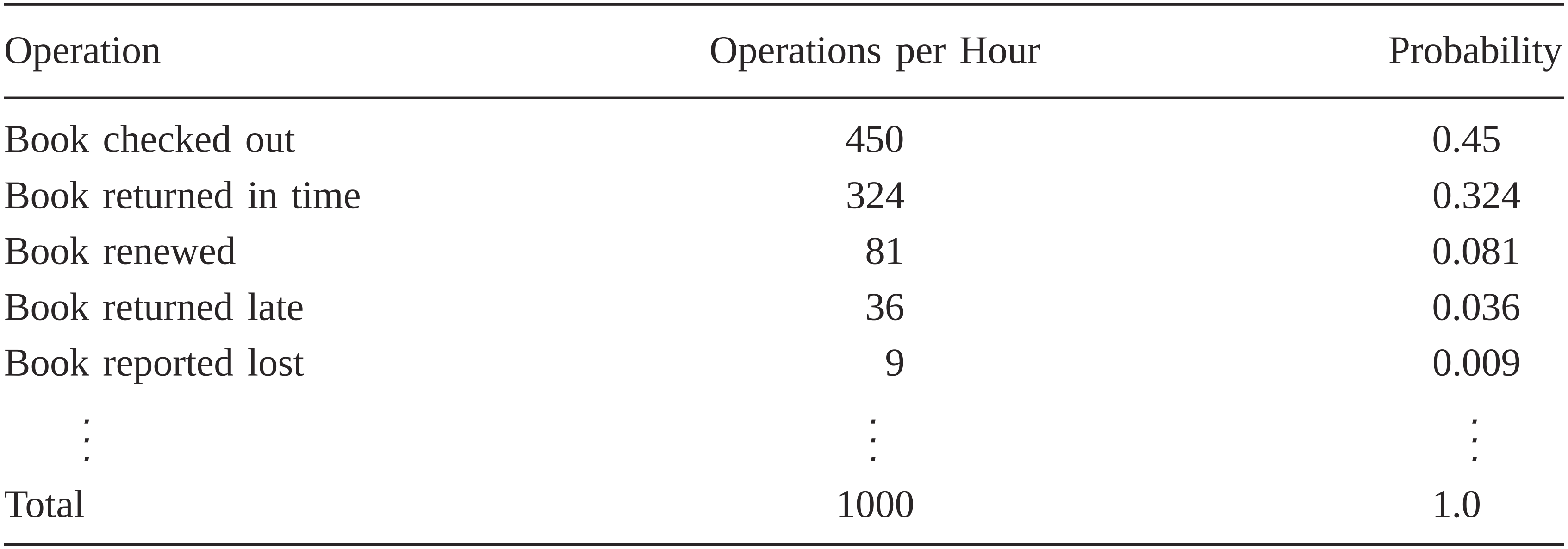
Operational Profiles

- Operational profile or usage profile
  - Describe how actual users operate a system
- Estimate the reliability of a system depends essentially on how it will actually be used in the field
- Representation of Operational Profile
  - Set of operations and its probability of occurrence
  - Ex: 3 operations A, B, C with probability of occurrence: 50, 30 and 2%
  - Operational profile: {(A, 0.5), (B, 0.3), (C, 0.02)}
- Two ways to represent:
  - Tabular representation
  - Graphical representation

Tabular representation of Operational Profile

- Tabular form with 3 columns
- Ex: Library Information System
  - 1st column: names of operations
  - 2nd column: the frequency of using the operation
  - 3rd column: the probability of using the operation

Example



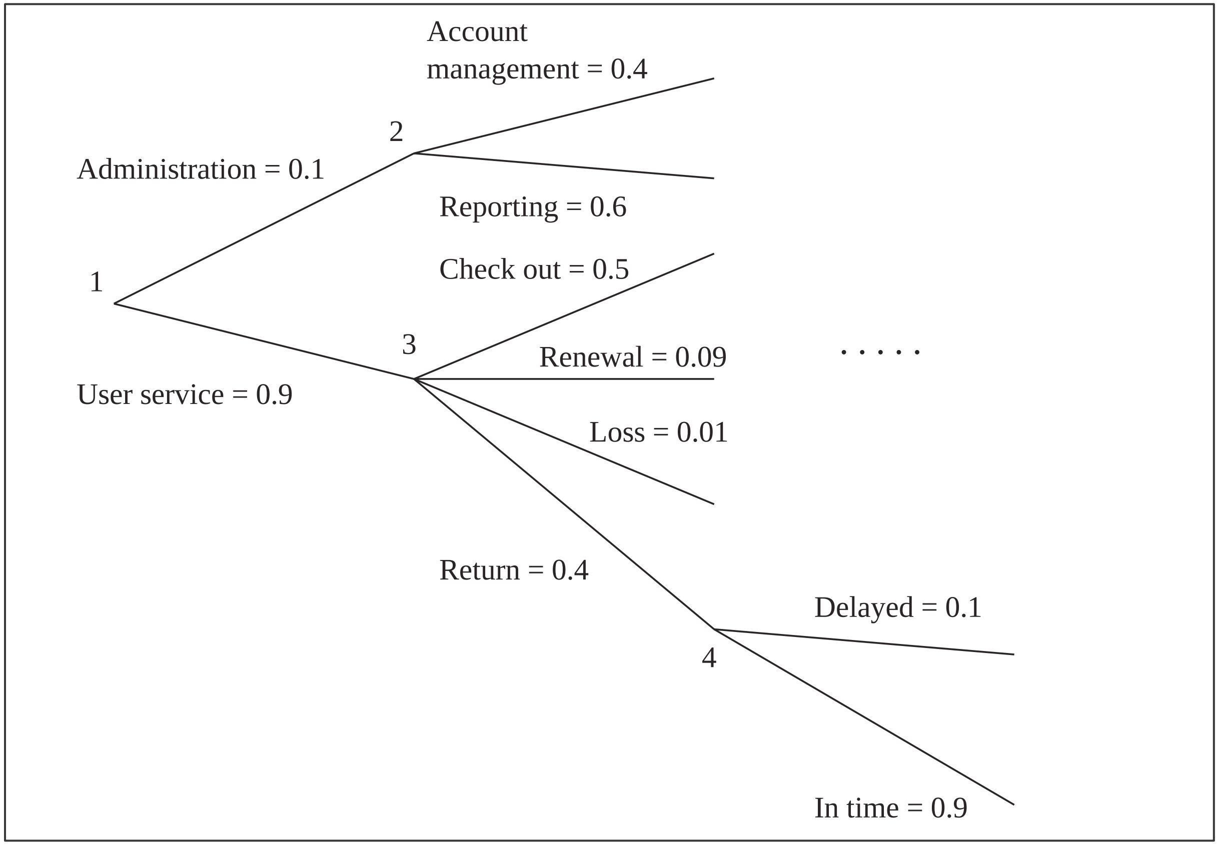
Operation	Operations per Hour	Probability
Book checked out	450	0.45
Book returned in time	324	0.324
Book renewed	81	0.081
Book returned late	36	0.036
Book reported lost	9	0.009
⋮	⋮	⋮
Total	1000	1.0

Example of Operational Profile of Library Information System

Graphical representation of Operational Profile

- Tree structure consisting of nodes and branches
  - Nodes represent attributes of operations
  - Branches represent values of attributes with the associated probability of occurrence
- Example: Library Information System

Example



Graphical Representation of Operational Profile of Library Information System

Which form to choose?

Thank you for your attention!!!