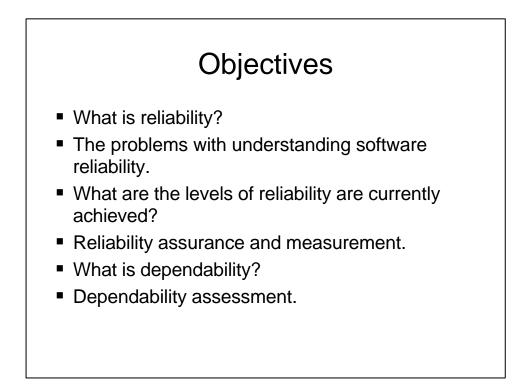
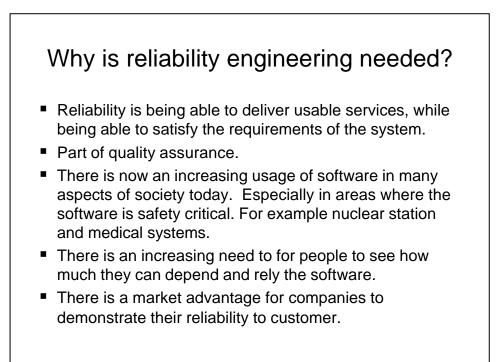
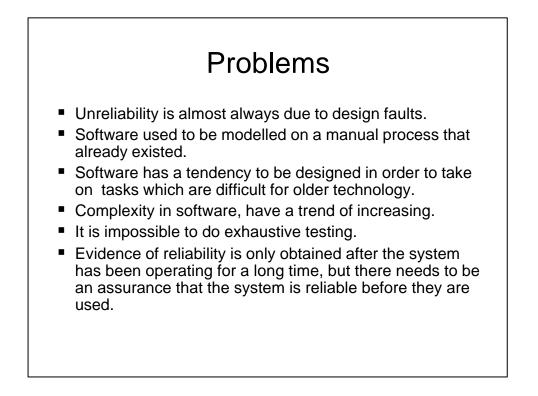
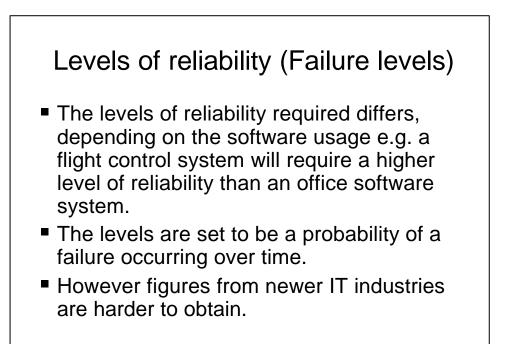
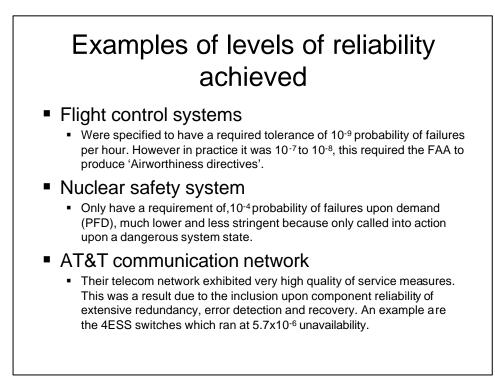
# Software Reliability and Dependability © Christine Loke & Simon Tolley











### Measurement and Assurance of Reliability 1

#### Testing under operational conditions

- Reliability growth modelling
  - Simulate the software's operational use noting times when failures occur and statically analyzing the data. The faults will then be removed.
  - Limitations
    - It is often difficult to create a test suite that is representative of operational use.
    - This model assumes successful fault removal, however it is possible that removing this fault will introduce new faults.
    - The data collected is quite limited and in order to demonstrate reliability levels the number of tests that would need to be executed to show a confidence in the PFD would be infeasible for a high level of reliability piece of software.

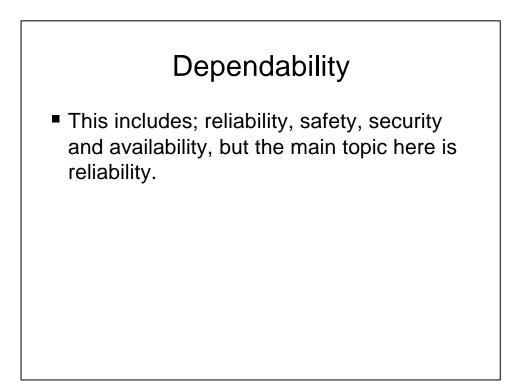
## Measurement and Assurance of Reliability 2

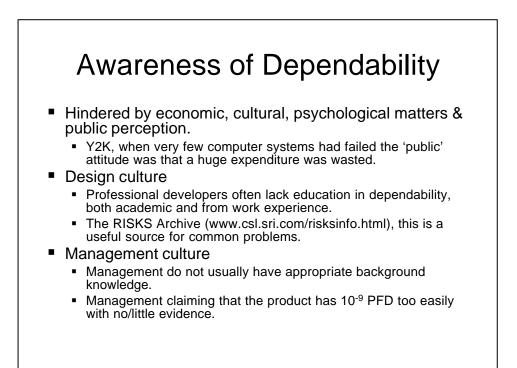
- Static Analysis of the software
  - A formal proof or analysis that shows that a class of fault is not present in the software.
  - Limitations
    - The analysis tool can only be used, as long as the complexity of the software is not too great.
    - There is the question of measuring the increase in confidence we have in the software.
    - For example; Malapas Analysis
      - The safety system software of the Sizewell nuclear reactor, showed some problems but they were claimed to have no safety implications. However certain parts of the system due to their complexity defeated the analytic tool.

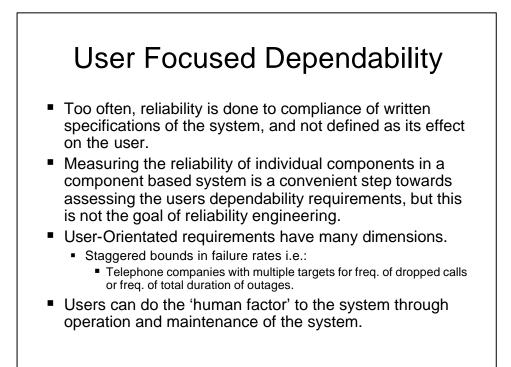
# Measurement and Assurance of Reliability 3

#### Structural model of reliability

- The reliability of the various components of the system is know and therefore the overall system reliability is estimated from the components reliability.
- This could also be used when Off the shelve components are used, and the reliability in previous systems are used to calculate the overall system reliability.
- Limitations
  - The data used for obtaining the reliability of the components are often inaccurate and not sufficient in detail.
    - There needs to be a simple model to capture interaction between the components in order for it to be manageable.
    - The difficulty in estimating the effect of redundancy, due to the order complexity.
  - The reliability calculated previously for components, may not reflect the reliability of the context usage.

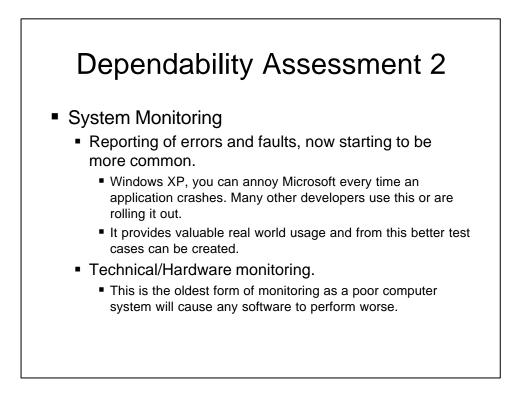






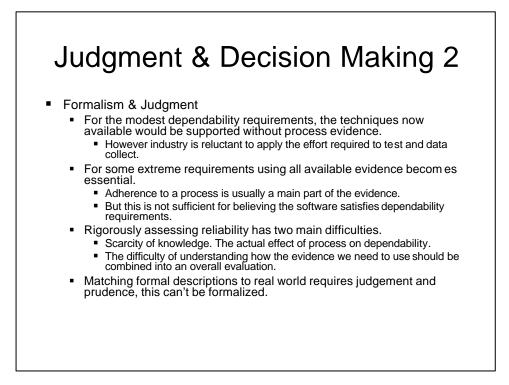
### **Dependability Assessment 1**

- Validation
  - Design cultures often ignore the need for validation.
- Failure Prevention
  - Proof that certain event can't happen.
  - Separation of sub-systems, memory protection prevents interference and propagation between different processes.
    - Modern airliners require guaranteed separation in the powerful software services that fly/run them.



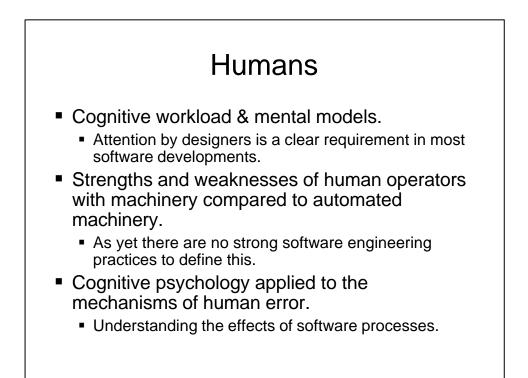
#### **Judgment & Decision Making 1**

- Engineering Approach
  - Often industry can't afford the extra effort on reliability. This view is limiting.
  - In achieving dependability, matching means to dependability targets is required.
  - Rational decisions must be taken taking into the affecting factors.
    - Overkill is a sensible strategy if the degree of cost of failure out ways the cost of overkill.
- Choice of Process
  - Most proposed methods for improving software reliability have intuitive appeal.
    - But measuring their whether the advantages are real and the cost is justified is seldom attempted.
  - The exploiting of available data to extract useful conclusions without overgeneralization poses demanding analytical problems.
  - However there is the need to base decisions on general laws that determine effectiveness of methods rather than on raw data.





- Often built using unplanned aggregation of computer-based systems. This creates new concerns.
  - Modelling complex systems without losing the ability to understand and solve the models.
  - Avoid the risk factors which previously negligible.
  - Detailed models are useless.
- Practices are often browed form other real world large-scale systems.
  - Communication/electrical distribution networks.
  - Modelling the evolution/propagation of disease breakouts.
- Grow without a general design.
- Emerging properties determine the user observed dependability.
- Not limited to the software side.
  - Economic, legal, environment & technological factors/developments.



### Reference

- Software Reliability Guidebook.
  - Robert L. Glass (1979).
- Software Reliability and Dependability: a Roadmap.
  - Bev Littlewood & Lorenzo Strigini

