

Critical Systems Specification

CIS/CSE 583



Types of Requirements

- Functional
 - Error checking, recovery
 - Protection against system failures
- Non-functional
 - Reliability, availability
- "Shall not"
 - Proscribe unsafe or insecure behaviors
 - Can sometimes be expressed as functional requirements



Factors in Computer-based System Reliability

- Hardware reliability
 - What is the probability of a hardware component failing, and for how long?
- Software reliability
 - How likely is it that software will produce bogus results? Software doesn't wear out.
- Operator reliability
 - How likely is it that the human operator will make an error?



Linkages

- Hardware errors can trigger unexpected signals or input to software
- Software can behave in unexpected ways
- Strange behavior confuses operator
- Confused, stressed operator makes mistake in handling situation
- Mistaken reaction further destabilizes the system



Reliability Metrics I

- Probability of Failure on Demand (POFOD): likelihood the system will fail when a request for service is made. A POFOD of 0.001 means that 1 in 1000 requests will fail.
- Rate of Failure Occurrence (ROCOF): likely frequency of occurrence for unexpected behavior. ROCOF of 2/100 means 2 failures in 100 time units (also called failure intensity)



Reliability Metrics II

- Mean time to failure (MTTF): the average time between system failures. An MTTF of 500 means that we expect one failure every 500 units.
- Availability (AVAIL): probability that a system will be available for use at a given time. An AVAIL of 0.998 means that for any 1000 time units, the system is likely to be available for 998 of them.



Meaning of “Time” in Metrics

- Time might be calendar time, processor time, or discrete units such as transactions
- Systems with continuous load—calendar time is fine
- Systems idle most of the time—processor time is better
- Transaction processing systems with variable-demand load (reservation systems or ATMs) are more concerned with ROCOF and might measure time units as transactions



Measurements

- Measure number of system failures over a large batch of requests to determine POFOD.
- Time or number of transactions between observed failures, for ROCOF and MTTF.
- Elapsed repair/restart time once failure occurs. This affects AVAIL.




Useless Non-functional Requirements

- “The software shall be reliable under normal use”
 - What does that mean? How will we measure it?
- “At most N faults per 1000 lines of code.”
 - If we can’t tell when all faults have been discovered, how can we know this?
 - Remember that failures are faults in action, and we’re measuring failures




Steps in Establishing Reliability Specification I

- For each sub-system, identify the different types of possible system failure, and the consequences of these failures
- From that analysis, classify the failures into appropriate classes (see next slide)




Failure Classes

- Transient/Permanent
 - T: occurs only with certain inputs
 - P: occurs with all inputs
- Unrecoverable/recoverable
 - System does (R: does not) require operator intervention to recover
- Corrupting/Non-Corrupting
 - Failure does (N: does not) corrupt system state or data



Steps in Establishing Reliability Specification II

- For each class define the reliability requirement using an appropriate metric.
 - Unrecoverable \Rightarrow PODOF
 - Recoverable \Rightarrow ROCOF
- Identify functional reliability requirements to reduce the probability of critical failures, where appropriate.



Example: ATM

- Machine used 300 times/day
- Machine lifetime: 8 years
- Software upgraded every 2 years
- 1000 machines in network
- $300 / \text{day} * 730 \text{ days} \approx 200,000$ transactions per software release per machine.
- 300,000 trans. per day
- Approx. 100,000,000 transactions on central database per year



Broad Classes of Failures

- Per-machine failures
- Central database failure
- Clearly, per-machine failures are more acceptable than system-wide failures



Examples of Failures

Permanent, non-corrupt.	System fails to read any input card; software restart required	ROCOF, 1/1000 days
Transient, non-corrupt.	Mag stripe can't be read from an undamaged card	ROCOF, 1/1000 trans.
Transient, corrupt	Pattern of trans. on the network corrupts database	Never (define as 1/200,000,000,000 transactions)



Note on Testing

- Note that it is not feasible to test the last requirement
- Suppose we simulate the network and it takes 1 second for a transaction
- 300,000 seconds to simulate one day
- But that itself is 3.5 days!



Safety Specification

- Do no harm (misquoting of the Hippocratic oath)
- See overhead



Hazard and Risk Analysis

