Non-functional requirements
Metrics
Reliability

Paul Jackson

School of Informatics
University of Edinburgh

NFRs or the "ities"

- Ways the system needs to be related to other systems and versions of itself:
  flexibility, maintainability, reusability, portability
- Things the system must not do or allow to be done
  security, integrity
- Properties of the system in use
  usability
  dependability, reliability, availability, robustness
  scalability
  performance, efficiency

etc.

NFRs in the development process

Must be identified along with functional requirements – at the end is too late. (Requires special care in agile developments).

Often tied up with architectural decisions: hard to modify late. For example, if you choose the AppFuse framework for your system, you are locked into its security model.

How much of an ity is needed? Often essential to

- quantify the requirement
- have a way to measure the system – “metrics”.

We’ll look at metrics generally and then at reliability metrics in particular.

Useful metrics should ideally be...

- Measurable – e.g. not someone's opinion of how complex something was
- Specified with a precision – i.e. a range in which measured values have to fall
- MEANINGFUL!! – there must be some reason to believe that numbers for the metric have something to do with something we care about!

Reliability

Reliability is a key non-functional requirement in many systems. But how does one specify reliability?

Several ways – most appropriate depends on the nature of the system.

Lets look at some.

POFOD

Probability of failure on demand is the probability that the system will fail when service is requested.

- Mainly useful for systems that provide emergency or safety services.
- E.g. the emergency shutdown in a nuclear power plant will never be used – but if it is, it shouldn't fail.
    - 'New' Sizewell B Primary Protection System specified 0.0001 – and achieved 0.001 in testing.

How to evaluate? Repeated tests in simulation. Expensive?

ROCOF

Rate of occurrence of failure is the number per unit time of failures (unexpected behaviour). 'Time' may mean elapsed time, processing time, number of transactions, etc.

- Mainly used for systems providing regular service, where failure is significant.
- E.g. banking systems.
  - VisaNet processes over 10^8 transactions/day. Failure rate is not published, but probably (much) less than 10^{-5} failures/transaction.

MTTF & MTBF

MTTF = Mean time to failure

MTBF = Mean time between failures

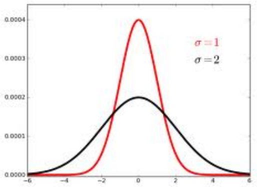
- Both mainly used where a single client uses the system for a long time. E.g. CAD systems – or indeed desktop PCs.
- MTTF used when system is non-repairable.
  - Popular metric for hardware components.
- MTBF used when system can recover from failures.
  - E.g. used for OS crashes

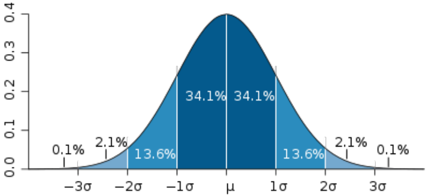
Q: What's the difference between MTBF and ROCOF?

Q: You buy a hard drive with an MTTF of 5 years. When will you replace it?

Statistics Digression

You know MTTF. Can you rely on that as an estimate?





Availability

Availability is the proportion of the time that the system is ‘available for use’. Often quoted as ‘five nines’, meaning 0.99999, ‘four nines’ etc.

Appropriate for systems offering a continuous service, where customers expect it to be there all the time. (‘Five nines’ is achieved by large data processing systems (e.g. VisaNet) – running on IBM mainframes, not PCs!)

Q: What’s the difference between availability and ROCOF?

Metrics Summary

Metric	**Appropriate when...**	**Example**
POFOD	System is rarely used	Airliner oxygen masks
ROCOF	Regular service, failure significant	Online site, correct customer charge
MTTF	Single client, extended use	disks
MTBF		desktop OS crash
Availability	Continuous service	online site up