

CSC 5524 : Software quality, metrics, tests, processes

J Paul Gibson, D311

`paul.gibson@telecom-sudparis.eu`

`http://www-public.telecom-sudparis.eu/~gibson/Teaching/CSC5524/`

Software Quality

`/~gibson/Teaching/CSC5524/CSC5524-SoftwareQuality.pdf`

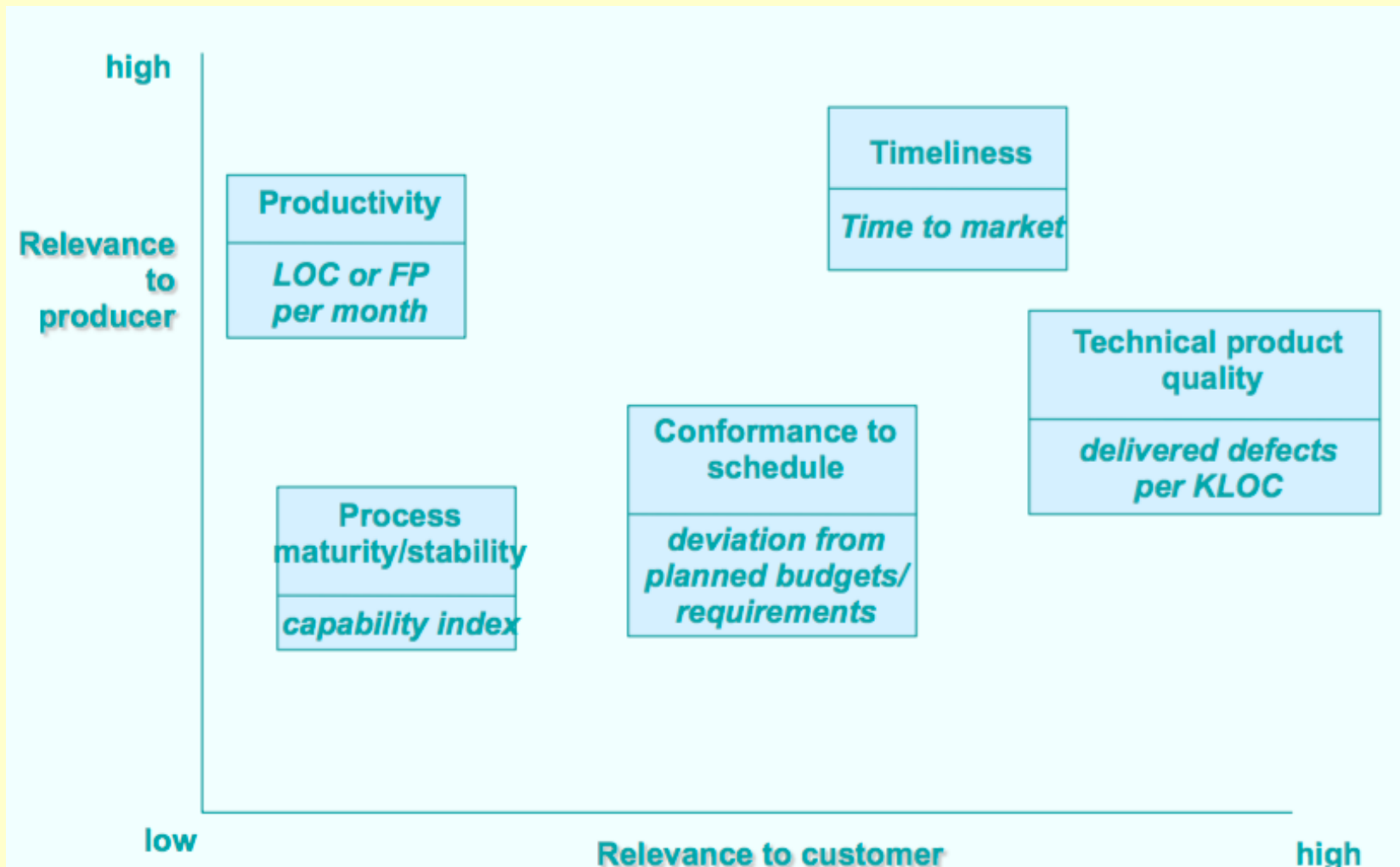
What is software quality?

1. Fitness for purpose?
2. Conformance to specification?
3. Absence of defects?
4. Degree of excellence?
5. Timeliness?
- 6....?

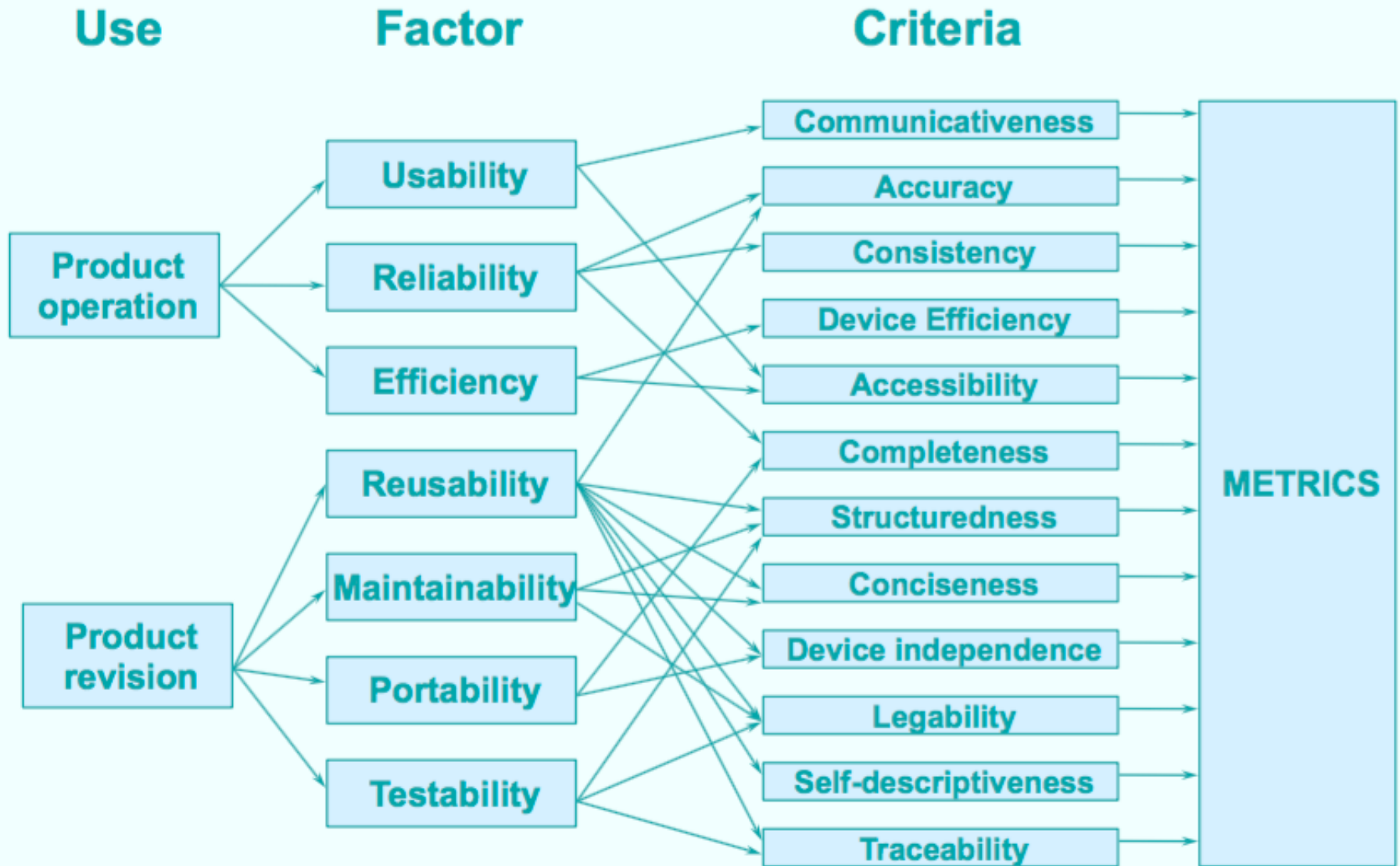
In the PBL problems, which criteria did you use?

Let's look at the quality of your solutions

Relevance



Models



Definition of system reliability

The reliability of a system is the probability that the system will execute without failure in a given environment for a given period of time.

Implications:

- 1.No single reliability number for a given system - dependent on how the system is used
- 2.Use probability to express our uncertainty
- 3.Time dependent

What is a software failure?

Alternative views:

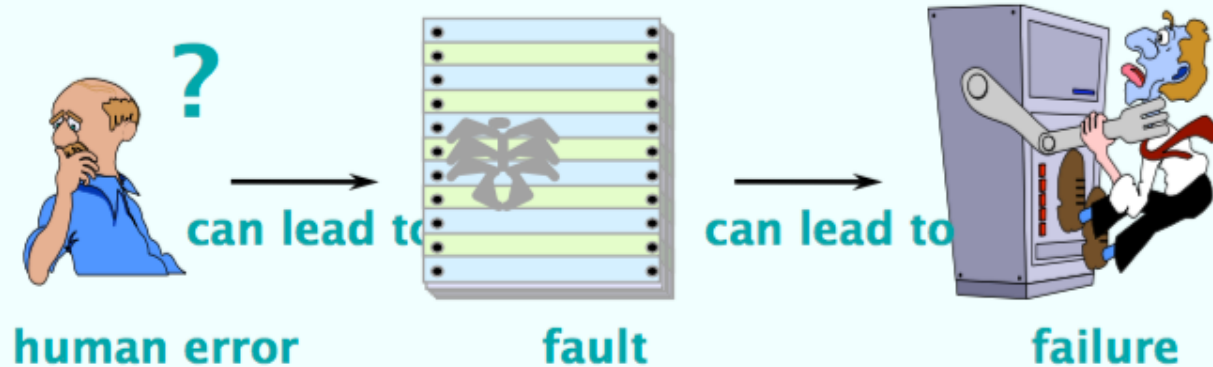
Formal view

Any deviation from specified program behaviour is a failure
Conformance with specification is all that matters
This is the view adopted in computer science

Engineering view

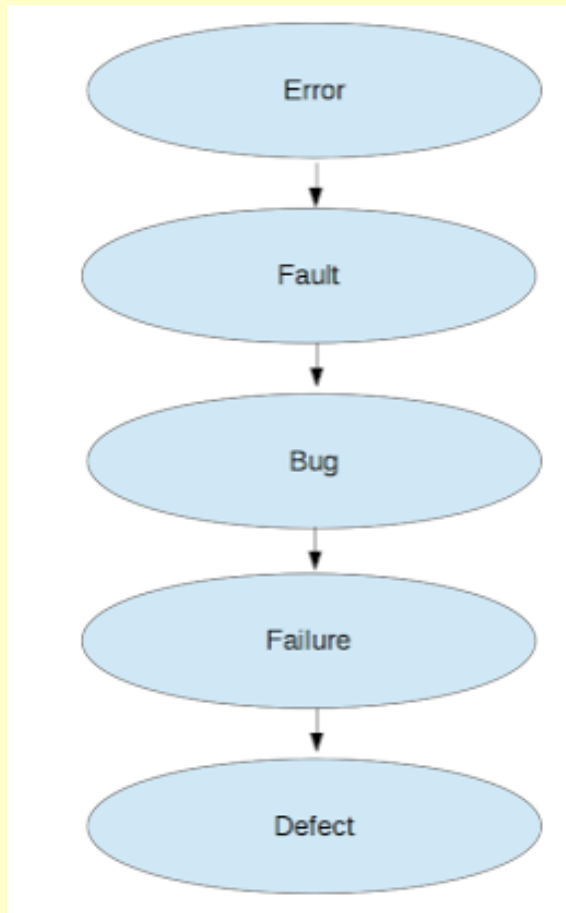
Any deviation from required, specified or expected behaviour is a failure
If an input is unspecified the program should produce a “sensible” output appropriate for the circumstances
This is the view adopted in dependability

Human errors, faults, and failures



- **Human Error**: Designer's mistake
- **Fault**: Encoding of an error into a software document/product
- **Failure**: Deviation of the software system from specified or expected behaviour

Difference between error, fault, bug, failure and defect



What is an **error**?

Error is deviation from actual and expected value.
It represents mistake made by people.

What is a **fault**?

Fault is incorrect step, process or data definition in a computer program which causes the program to behave in an unintended or unanticipated manner.
It is the result of the error.

What is a **bug**?

Bug is a fault in the program which causes the program to behave in an unintended or unanticipated manner.
It is an evidence of fault in the program.

What is a **failure**?

Failure is the inability of a system or a component to perform its required functions within specified performance requirements.
Failure occurs when fault executes.

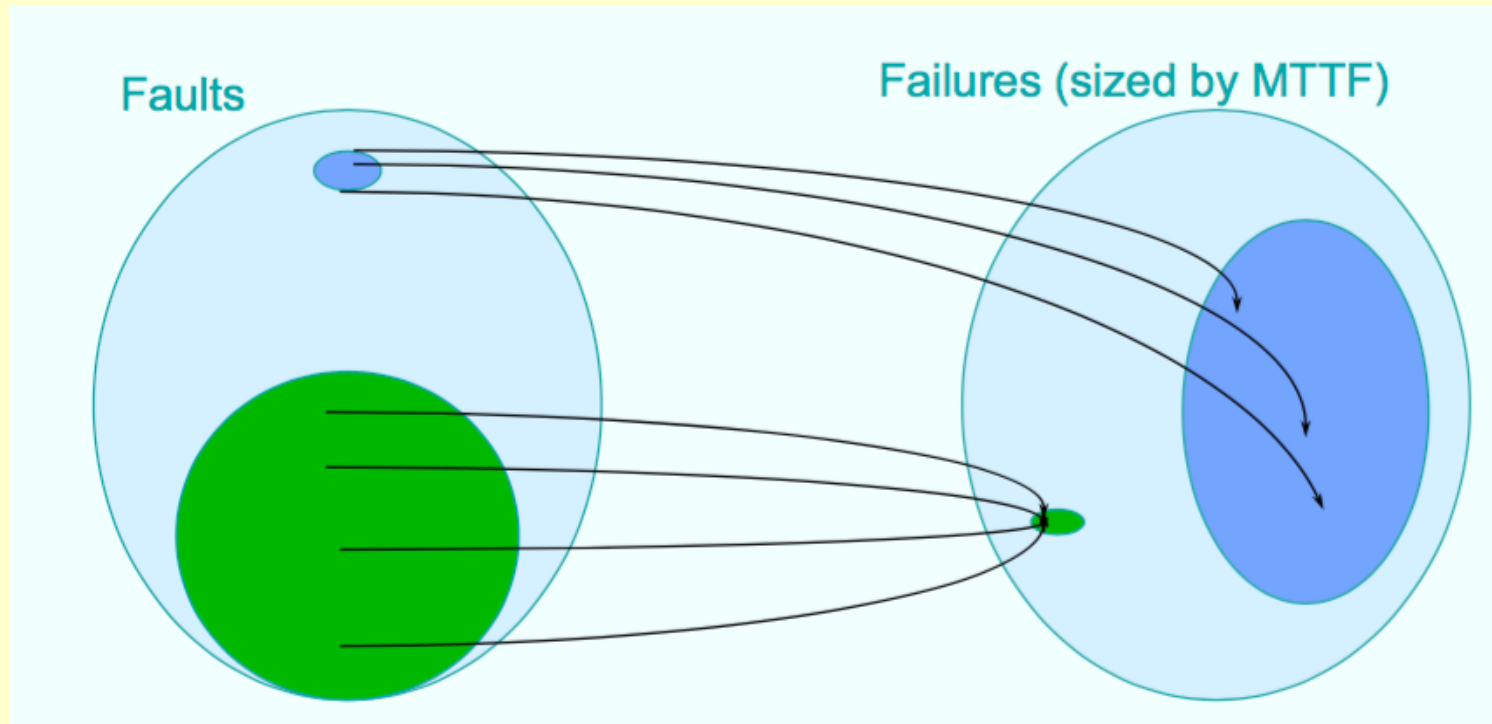
What is a **defect**?

A defect is an error in coding or logic that causes a program to malfunction or to produce incorrect/unexpected results.
A defect is said to be detected when a failure is observed.

One suggestion, from:

<http://softwaretestingbykunti.blogspot.fr/2012/11/difference-between-error-fault-bug.html>

Relationship between faults and failures

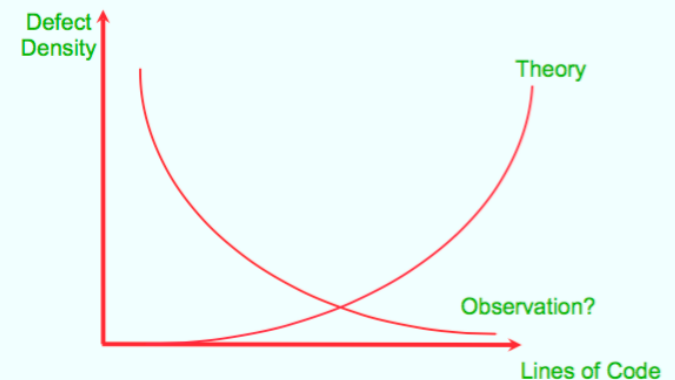


1. Most faults are benign
2. For most faults: removal will not lead to greatly improved reliability
3. Large reliability improvements only come when we eliminate the small proportion of faults which lead to the more frequent failures
4. Does not mean we should stop looking for faults, but warns us to be careful about equating fault counts with reliability

The 'defect density' measure: an important health warning

- Defects = {faults} \cup {failures}
 - ◆ but sometimes defects = {faults} or defects = {failures}
- **System defect density** = $\frac{\text{number of defects found}}{\text{system size}}$
 - ◆ where size is usually measured as thousands of lines of code (KLOC)
- Defect density is used as a de-facto measure of software quality.

Defect density vs module size



A study in relative efficiency of testing methods

Testing Type	Defects found per hour
Regular use	0.21
Black box	0.282
White box	0.322
Reading/Inspections	1.057

Tracking incidents to components

Incidents need to be traceable to identifiable components - but at what level of granularity?

1. Unit
2. Module
3. Subsystem
4. System

Generic incident data collection

- *What*: Product details
- *Where* (Location): Where is it?
- *Who*: Who found it?
- *When* (Timing): When did it occur?
- *What* happened (End Result): What was observed?
- *How* (Trigger): How did it arise?
- *Why* (Cause): Why did it occur?
- *Severity/Criticality/Urgency*
- *Change*

Software Quality is very tightly linked to:

- testing quality
- metrics/measurements
- software process quality

We will continue with some more problems as in introduction to these topics