

### 8.3. Testarea sistematica si reproductibila

#### 8.3.1. Caracteristicile testarii

How we design the software affects testability

- **Operability** – *The better it works, the more efficiently it can be tested.*
  - Bugs add overhead of analysis and reporting to testing.
  - No bugs block the execution of the tests.
  - The product evolves in functional stages (allowing concurrent testing)
- **Observability** – *What you see is what you test.*
  - A distinct output is generated for each input
  - System state and variables should be visible or queriable during execution (past states and variables too)
  - Incorrect output is easily identified
  - Internal errors are detected through self-testing, and are automatically reported
  - Source code is accessible



UNIUNEA EUROPEANĂ



MINISTERUL ÎNȘĂȘĂRII ȘI  
PROIECTĂRII



FONDUL SOCIAL EUROPEAN  
2007-2013



INSTRUMENTE STRUCTURALE  
2007-2013

### 8.3. Testarea sistematica si reproductibila

#### 8.3.1. Caracteristicile testarii

- **Controllability** – *The better we can control the software, the more testing can be automated and optimized.*
  - All possible outputs can be generated through some combination of inputs
  - All code is executable through some combination of input
  - Software and hardware states can be controlled directly by the test engineer
  - Input and output formats are consistent and structured
  - Tests can be conveniently specified, automated and reproduced
- **Decomposability** – *By controlling the scope of testing, we can more quickly isolate problems and perform smarter retesting*
  - Software is built from independent modules
  - Modules can be tested independently
- **Simplicity** – *The less there is to test, the more quickly we can test it.*
  - Functional simplicity – no extra features beyond requirements
  - Structural simplicity – partition architecture to minimize the propagation of faults
  - Code simplicity