

### 8.1. Notiuni generale despre testare

#### 8.1.1. Scop

**Testarea** – este un proces centrat in jurul scopului de a gasi defecte intr-un sistem. Fie ca e vorba de depanarea sistemului sau de verificarea unor conditii de acceptare, procesul de identificare a defectelor este o parte esentiala a procesului de testare. Testarea reprezinta procesul de imbunatatire a calitatii unui sistem.

Nu in ultimul rand, procesul de testare furnizeaza informatii organizatiei despre modul in care este produs sistemul. Testarea, in sine, nu imbunatateste in mod direct calitatea sistemului ci, mai degraba, indica slabiciunile acestuia si permite componentei de management sa aloce resurse pentru imbunatatirea calitatii.

Pentru a indeplini aceste functii, fiecare proces de testare trebuie prevazut cu un plan de testare (calendar de activitati) care sa specifice ce trebuie testat si cu un set de teste care trebuie executate. Trebuie subliniat faptul ca este imposibila producerea unui sistem fara defecte datorita resurselor limitate. Scopul testarii este acela de a indica modul in care resursele existente pot fi utilizate pentru obtinerea unei cresteri maxime a calitatii.

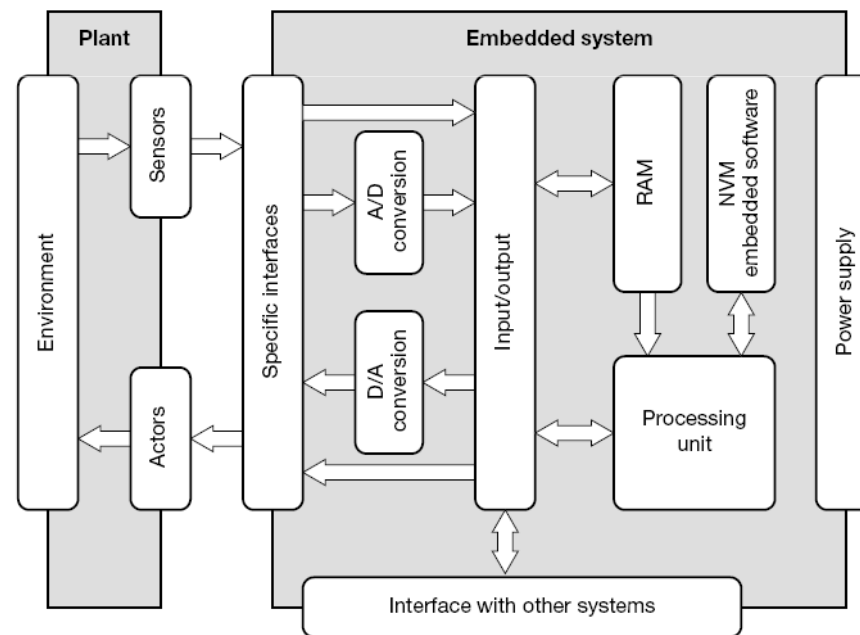
Elemente cheie: *lifecycle, testing process, testing unit, infrastructure.*

### 8.1. Notiuni generale despre testare

#### 8.1.2. Testarea sistemelor embedded

##### Testarea sistemelor embedded:

- Proces axat pe functionalitatea sistemului embedded;
  - Proces axat pe componentele sistemului embedded;
  - Proces axat pe caraterul real-time al sistemului embedded (soft-, hard-);
  - Proces axat pe destinatia sistemului embedded (biomedical, aerospace, military, long-term, etc.);
- => Nu exista o abordare **unica** pentru orice sistem embedded.

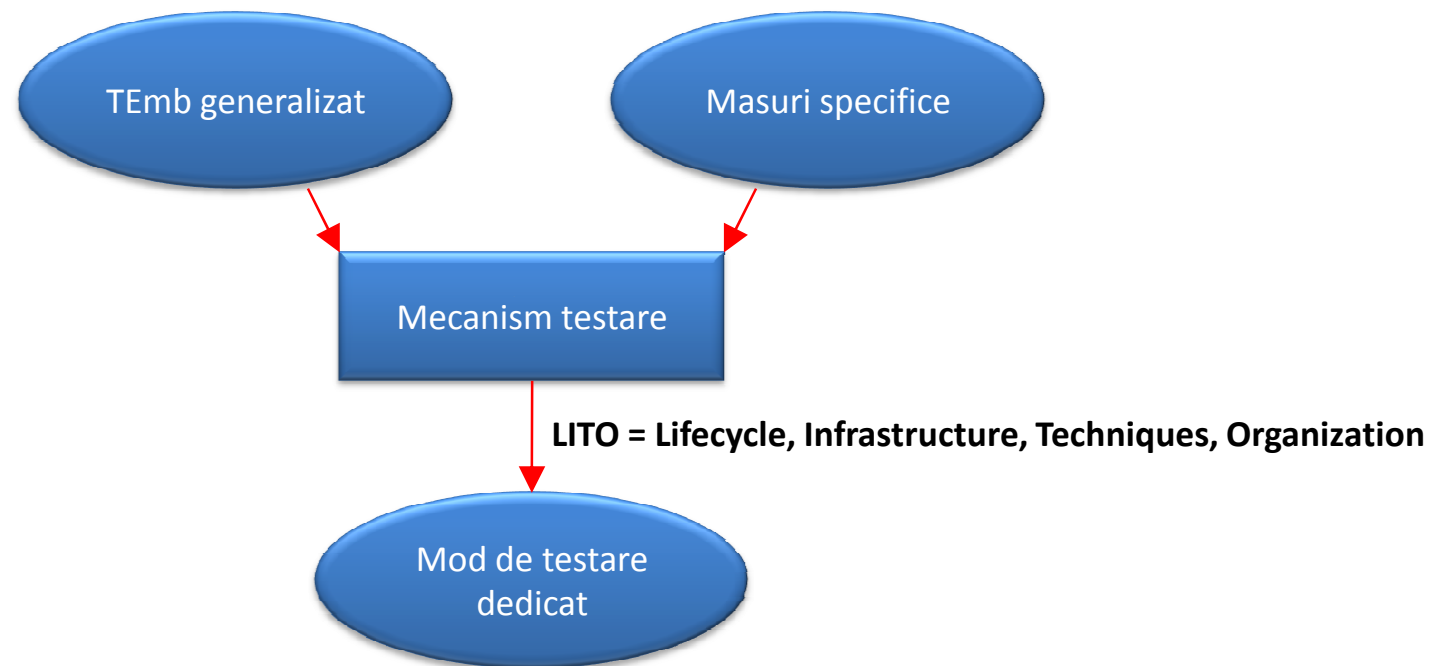


### 8.1. Notiuni generale despre testare

#### 8.1.3. Metoda TEmb

**TEmb:**

- Este e o metoda care permite abordarea unui mod de testare potrivit pentru un sistem embedded particular;
- Propune un mecanism de asmlare a unor elemente generice aplicabil oricarui proces de test;
- Propune un set de masuri specifice caracteristicilor relevante ale sistemului embedded;



### 8.1. Notiuni generale despre testare

#### 8.1.3. Metoda TEmb

**Mecanismul** care se afla la baza generarii procesului dedicat de testare va analiza urmatoorii factori:

**Riscul:** masurile de testare trebuie sa acopere riscul implicat de calitatea necorespunzatoare a produsului.

**Caracteristicile sistemului:** masurile de testare trebuie sa ia in calcul problemele ridicate de caracteristicile tehnice ale sistemului analizat: algoritmi, semnale AD, safety-critical, etc.

Characteristics	Lifecycle	Infrastructure	Techniques	Organization
Safety critical	Master test MOD-00-56, safety test, load/stress level	Coverage analysis	FMEA = Failure Mode Effect Analysis FTA = Fault Tree Analysis Rare-event test	Safety manager Safety engineer
Technical- scientific algorithms	Algorithm validation	Coverage analyzers, Threat detectors	Evolutionary algorithms, Threat detection	Math expertise

### 8.1. Notiuni generale despre testare

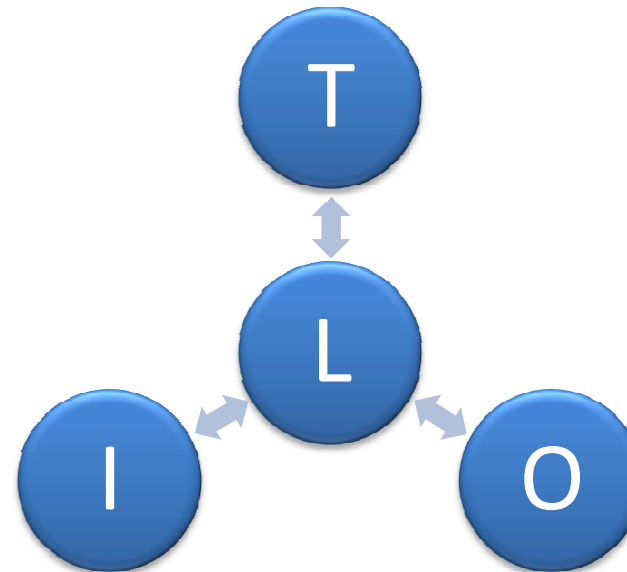
#### 8.1.3. Metoda TEmb

**Lifecycle:** ciclul de viata al produsului – defineste ce activitati trebuie efectuate si in ce ordine. Permite factorilor de management si de testare sa aiba control asupra intregului proces;

**Infrastructure:** infrastructura defineste ce este necesar in mediul de testare pentru a putea indeplini activitatile planificate;

**Techniques:** set de tehnici standardizate pentru efectuarea testelor;

**Organization:** defineste rolurile si expertiza necesara pentru personalul de testare precum si modul de interactiune inter- si intra-compartiment;

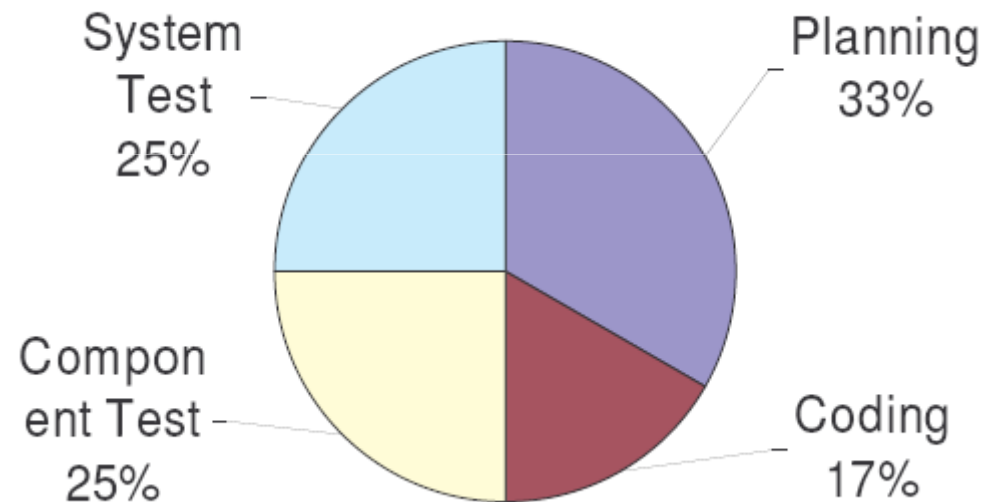


### 8.1. Notiuni generale despre testare

#### 8.1.4. Ciclul de viata al sistemului

**Lifecycle:** ciclul de viata al produsului – defineste ce activitati trebuie efectuate si in ce ordine. Permite factorilor de management si de testare sa aiba control asupra intregului proces;

Distributia timpului in proiecte software: accentuarea planificarii si testarii [Brooks]



*The sooner you start coding, the longer it will take to finish the program*