

# Dependabilitate & securitate IT&C

 Căutați pe acest site

[Home](#)
[Calitate & Fiabilitate](#)
[Fiabilitate & SEcuritate \(FISE\)](#)
[Testare](#)
[LABORATOR Atasamente](#)
[Sitemap](#)
[Home](#) > [Calitate & Fiabilitate](#) > [Notiuni de Fiabilitate](#) >

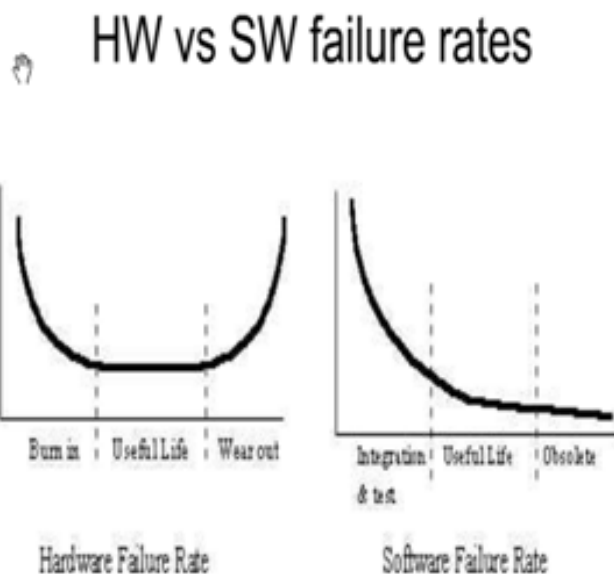
## 7. Fiabilitate HW vs SW



### DEFINIȚII

- **ANSI** Fiabilitatea software este definită ca probabilitatea unui produs software(SW) de a funcționa pt o perioadă specificată de timp într-un mediu precizat.
- **IEEE**
  - **Fiabilitatea** este abilitatea unui system de a-și îndeplini funcțiile încredințate pt o anumită perioadă de timp.
  - **Managementul Fiabilității Software** este procesul optimizării fiabilității software printr-un program care: previne erorilor SW, detectarea și eliminarea erorilor și utilizarea unor măsurători în condițiile existenței unor constrângeri de resurse, planificare și performanțe.

### Intesitatea de defectare HW vs SW



Grafic (l,t)

#### Caracteristici specifice produselor SW

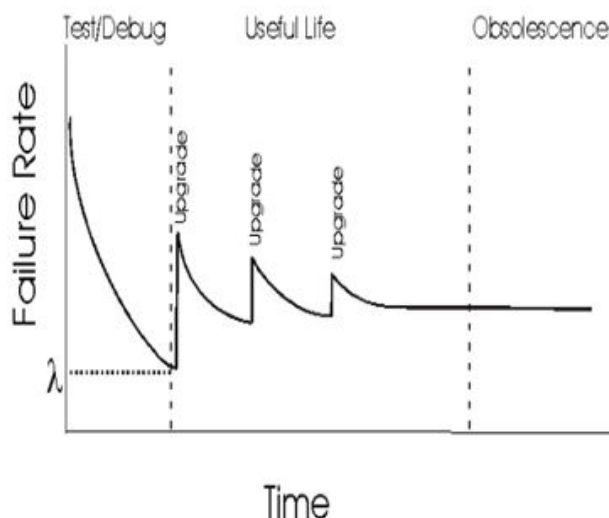
- **Cauza defectărilor** : defectările SW sunt numai defecte de proiectare.
- **Uzura** : SW nu îmbătrânește
- **Reparabilitatea sistem** : Restart periodic poate ajuta rezolvarea problemelor software
- **Ciclu de viață și dependența de timp**: funcția de fiabilitate nu este o funcție de timp de funcționare
- **Factorii de mediu** : nu influențează F SW, cu excepția situației în care afectează datele de intrare
- **Predicția fiabilității** : F SW nu poate fi estimată pe baze fizice, deoarece depinde în proiectare doar de factorul uman
- **Redundanța** : F SW nu poate fi îmbunătățită dacă sunt folosite mai multe componente identice
- **Interfețe** : interfețele SW sunt pur conceptuale cu excepția că sunt vizuale
- **Motivatori ai intensității de defectare**: de cele mai multe ori nu poate fi estimată din analize separate
- **Construirea SW cu componente standard** : componentele testate îndelung și bine cunoscute vor ajuta la creșterea Fiabilității și Mentenabilității F&M. În domeniul industriei SW nu se observă această tendință. Reutilizarea codului este o încercare dar destul de limitată. Pt SW în acest moment nu există componente standardizate cu excepția anumitor structuri logice.

#### F SW

- **Testare- depanare (upgrade)**
- **Viața utilă (upgrade)**
- **Uzură morală**

Grafic (l,t)

## Revised SW reliability



### Modele ale F SW

	Modele predictive	Modele estimative
Date folosite	Folosesc date istorice	Folosesc date din etapa de dezvoltare
Momentul folosirii in ciclul de viață	Uzual construite inaintea fazei de dezvoltare sau testare	Uzual construite mai târziu in ciclul de viață. Nefoloste in faza de dezvoltare
Timp	Predicția F pt momente de t ulerioare	Estimează F prezentă sau in viitor

### Tehnici utilizate in F SW

- **Tendința F** Inregistrarea datelor defecțiunilor SW pt a dezvolta un profil operațional al sistemului pt o perioadă specificată de timp
- **Fiabilitatea predictivă** atribuie probabilități profilului operațional a unui sistem SW. De ex sistem are 5% șanse de defectare in următoarele 60 h de funcționare

### Categorii de modele

- **Error Seeding**: estimează nr de erori intr-un program utilizând eşantioane la diferite momente. Erorile sunt împărțite in erori intrinseci și erori induse(injectate). Numărul de erori intrinseci este estimat din nr de erori induse și raportul erorilor obținute din etapa de depanare.
- **Intensitatea de defectare**: utilizată pt a studia nr de erori aparute intre doua momente ale defectării.
- **Curve Fitting**: utilizează analiza regresiiilor pt a studia corelația dintre complexitatea software și nr de defecțiuni din program, numărul de schimbări sau intensitatea de defectare
- **Creștere F**: măsoară și prezice creșterea de F in urma procesului de testare. Creșterea F reprezintă de asemenea F sau intensitatea de defectare a sistemului ca funcție de t sau nr cazurilor de test.

**METRICI**

- Contorizarea liniilor
  - **Linii de cod (LOC)**
  - **Linii de cod sursă (SLOC)**
- Complexitate și structură
  - Complexitate ciclomatică(CC)
    - Număr de căi independente pt a parcurge programul sursă
    - Număr de module
    - Nr de instrucțiuni GOTO
- Orientate obiect
  - Nr de clase
  - Ponderea metodelor pe Clasa(WMC)
  - Cuplarea între obiecte(CBO)
  - Răspunsuri pt o clasă (RFC)
  - Nr claselor copil(NOC)
  - Adâncimea arborelui de moștenire(DIT)

**Tendința dezvoltării SW**

## Software Size Trend

**Costuri**

## Cost

System	Cost (FY04\$)	Description (All involved loss of data)
Ariane 5 (1996)	\$594M	Software error shut down redundant inertial reference systems, resulting in loss of control and aerodynamic breakup.
Delta III (1998)	\$336M	Software error did not account for normal roll oscillation, resulting in loss of attitude control and auto-destruct.
Titan 4B (1999)	\$1.5B	Misplaced decimal point in Centaur flight software causes premature propellant depletion and deployment of payload in incorrect low orbit.
Mars Climate Orbiter (1999)	\$524M	Failure to use metric units in ground software trajectory models caused steeper than expected entry trajectory and destruction while entering the Martian atmosphere.
Zenit 3SL (2000)	\$367M	Software error caused premature second stage shutdown and the satellite failed to reach orbit.
Messenger (2004)	\$23.9M	Software testing and other factors caused launch delays and a new launch profile requiring an additional two years of cruise time to complete the mission – with partial loss of data.

## Procesul de dezvoltare SW

## SW Development Process

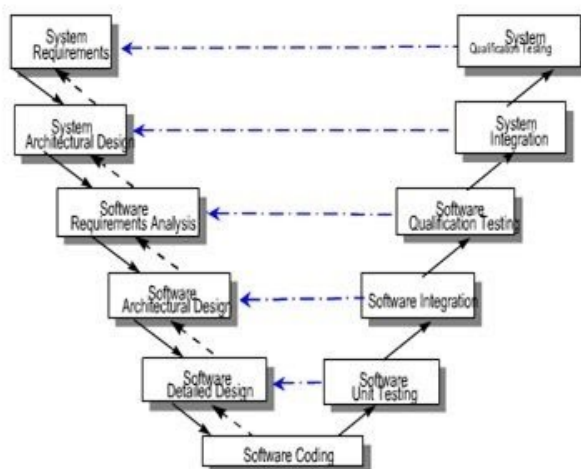


Fig.

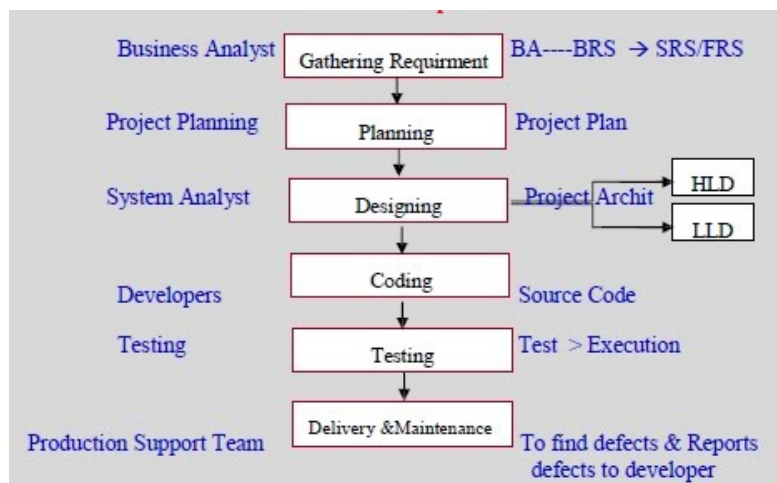
## TESTARE

- Cele mai utilizate metode de verificare și validare(V&V)

- Testare integrare vs unitate
- Simulare: de nivel mic către nivel înalt
- Avantaje
  - Ușor de înțeles
  - Scalabil
  - Captează toate tipurile de erori
- Dezavantaje
  - Nu e exhaustivă
  - Rezultate slabe pt sisteme cu amențări multiple
- Testare convențională
  - Cazuri de test și drivere
  - Teste la nivel unitate
  - Teste de integrare
  - Acoperire parțială a erorilor
- Analiza statică
  - Nu sunt necesare date de intrare și drivere
  - Verificare a codului sursă, compiler front end
  - Analiza fluxului de date și de control

## TESTARE SOFTWARE

SDLC



- **Eroare** (error)- orice acțiune umană care poate produce probleme in sistem
- **Defect** (defect) orice comportare diferită a sistemului față de comportarea așteptată
- **Cădere** (failure) orice comportament diferit identificat de un utilizator final

◦ Prezența erorilor implică apariția defectelor și prezența defectelor implică căderea produsului

◦ **Când apar defectele in SDLC ?**

1	2	3	4
Cerință corectă	Cerință corectă	Cerință corectă	Cerință incorectă
Proiectare conform cerinței	Proiectare conform cerinței	Greșeli in proiectare	Proiectare conform cerinței
Dezvoltare conform cerinței	Greșeli in dezvoltare	Dezvoltare conform cerinței	Dezvoltare conform cerinței
Produs corect	Produs conține defecte de codificare	Produs conține defecte de proiectare	Produs greșit

Din tabel rezultă importanța testării timpurii

**Testarea timpurie** : executarea de teste cât mai repede posibil in SDLC pentru depistarea defectelor cât mai repede. Testarea timpurie este utilă in reducerea costurilor eliminării defectelor.

De ce produsele software( SW) au defecte ?

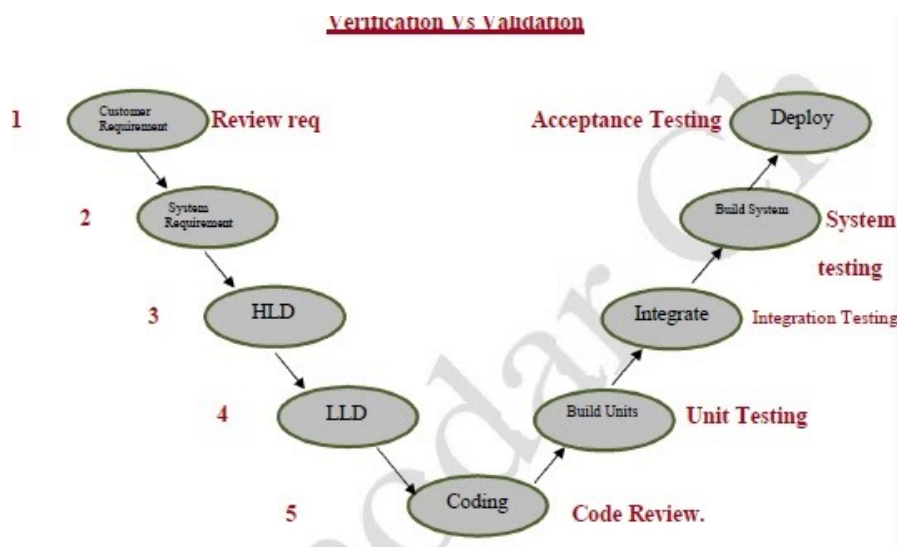
- Cerințe incorecte ;
- Proiectare greșită ;
- Codificare greșită ;
- Complexitate algoritmică ;
- Tehnologii complexe ;
- Presiunea lucrului ;
- Cerințe modificate frecvent

**TESTAREA** este un proces de verificare a faptului că realizăm produsul cerut și de asemenea că produsul realizat este corect.

**TESTAREA SW = VERIFICARE + VALIDARE**

VERIFICAREA este un proces prin care constatăm că am dezvoltat produsul dorit. Este cunoscută și ca **testare statică**.

VALIDAREA este un proces prin care constatăm că produsul face ceea ce ne-am propus. Este cunoscută și ca **testare dinamică**.



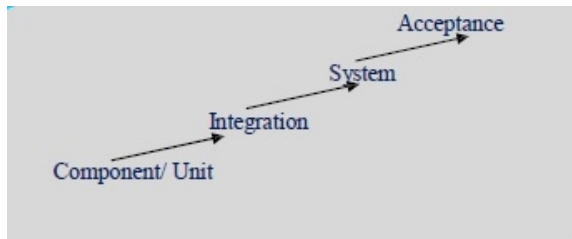
1,2,3,4,5 reprezintă procesul de verificare și cealaltă parte procesul de validare.



#### Tehnici de testare SW

- Testare statică ;
- Testare tip 'cutie albă'
- Testare tip 'cutie neagră'
- Testare tip 'cutie gri'

#### Nivele ale testării dinamice



#### VALIDARE

##### Metode de testare

Tradițional orientat spre succes