

## 11 stiluri de testare software [Kaner & Bach]

1. Testarea funcțiilor – *functional testing*
2. Testarea domeniilor de valori – *domain testing*
3. Testarea bazată pe specificații – *specification-based testing*
4. Testarea axată pe riscuri – *risk-based testing*
5. Testarea la limită – *stress testing*
6. Testarea de regresie – *regression testing*
7. Testarea de către utilizatori – *user testing*
8. Testarea bazată pe scenarii – *scenario testing*
9. Testarea bazată pe stări – *state-model based testing*
10. Testarea automată de volum ridicat – *high volume automated testing*
11. Testarea de explorare – *exploratory testing*

# 1. Testarea funcțiilor

- fiecare funcție este testată separat
- se testează funcționalitatea de bază
- teste ușoare, cu caracter general, nu foarte precise
- cerințe -> use cases -> teste funcționale
- Testare funcțională vs. Testare non-funcțională (ex.)

portabilitate, securitate, consum de resurse, încărcare

## 2. Testarea domeniilor de valori

- împărțirea în clase de echivalență
- alegerea reprezentanților
- valorile limită

### 3. Testarea bazată pe specificații

- un fel de testare funcțională
- parcurge amănunțit specificațiile -> cazuri de test
- specificații amănunțite -> valoare ridicată a testării
- specificații vagi -> valoare scăzută a testării

## 4. Testarea axată pe riscuri

- Testorul imaginează cazuri de test bazate pe risc (Ex.)
- Gruparea testelor în funcție de risc -> clasificare puternică
- Ușurință în crearea testelor, cu focusare pe un anumit risc
- Testorul gândește din perspectiva riscurilor, programatorul din perspectiva specificațiilor

## 5. Testarea la limită

- se urmărește depășirea limitelor impuse (Ex.)
- foarte importantă pentru aplicațiile online, multiuser (nr. utilizatori, nr. conexiuni DB)
- orice aplicație trebuie testată la limită (analogie cu testarea domeniului de valori) – se testează atât valorile limită specificate, cât și valorile ce depășesc aceste limite
- stabilește stabilitatea sistemului

## 6. Testarea de regresie

- necesită atenție deosebită în crearea testelor (manual vs. automat)
- refolosirea cazurilor de test în versiuni ulterioare
- documentație foarte atentă pentru mentenanță
- verifică dacă modificările aduse proiectului au influențat sau nu funcționalitatea conform specificațiilor

## 7. Testarea de către utilizatori

- utilizatori reali
- testarea: pe baza specificațiilor sau nu
- testele arată impactul asupra utilizatorilor finali – gradul de acceptanță a acestora
- majoritatea acestor teste vor fi simpliste



## 8. Testarea bazată pe scenarii

- Dezvoltă cazuri de test complexe – pe scenarii
- caracteristici: motivație, credibilitate, complexitate, ușor de evaluat
- Soap Operas [Hans Buwalda]

## 9. Testarea bazată pe stări

- automat cu stări finite
- teste automate bazate pe model
- caracteristici: credibil și motivant

## 10. Testarea automată de volum ridicat

- număr mare de teste
- rezultatele vor fi testate cu un *oracol*
- rețele sunt executate și interpretate de către calculator
- Testare automată vs. Testare asistată de calculator

## 11. Testarea de explorare

- proiectarea de noi teste pe baza rezultatelor obținute de testele existente
- învățare din experiența testelor anterioare
- explorare = experiență + creativitate + învățare