

# Testarea

0 imagine de ansamblu

# Cuprins

01

...

## Tipuri de testare

Care sunt tipurile de testare și când le folosim?

02

...

## Tehnici de testare

Care sunt tehnicile de testare și când le aplicăm?

03

...

## Nivele de testare

Unit, Integration, System, Acceptance

04

...

## Ciclul de dezvoltare software

Agile, Kanban & Scrum

05

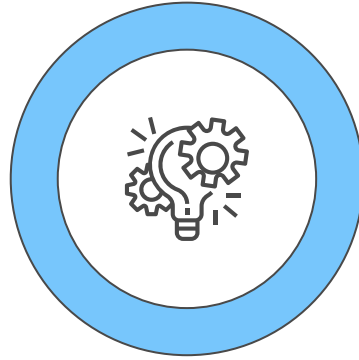
...

## Cele 7 principii de testare



# 01

## Tipuri de testare



# Testarea functionala

Testare efectuată pentru a evalua dacă o componentă sau un sistem satisface cerințele funcționale.  
- ISTQB 2018 Curriculum

...

# Exemple de testare functionala

Sanity testing - este un test de bază pentru a evalua rapid dacă o afirmație sau rezultatul unui calcul poate fi adevărat.

Smoke testing - este o testare preliminară pentru a dezvălui erori simple suficient de severe pentru, de exemplu, a respinge o viitoare lansare de software.

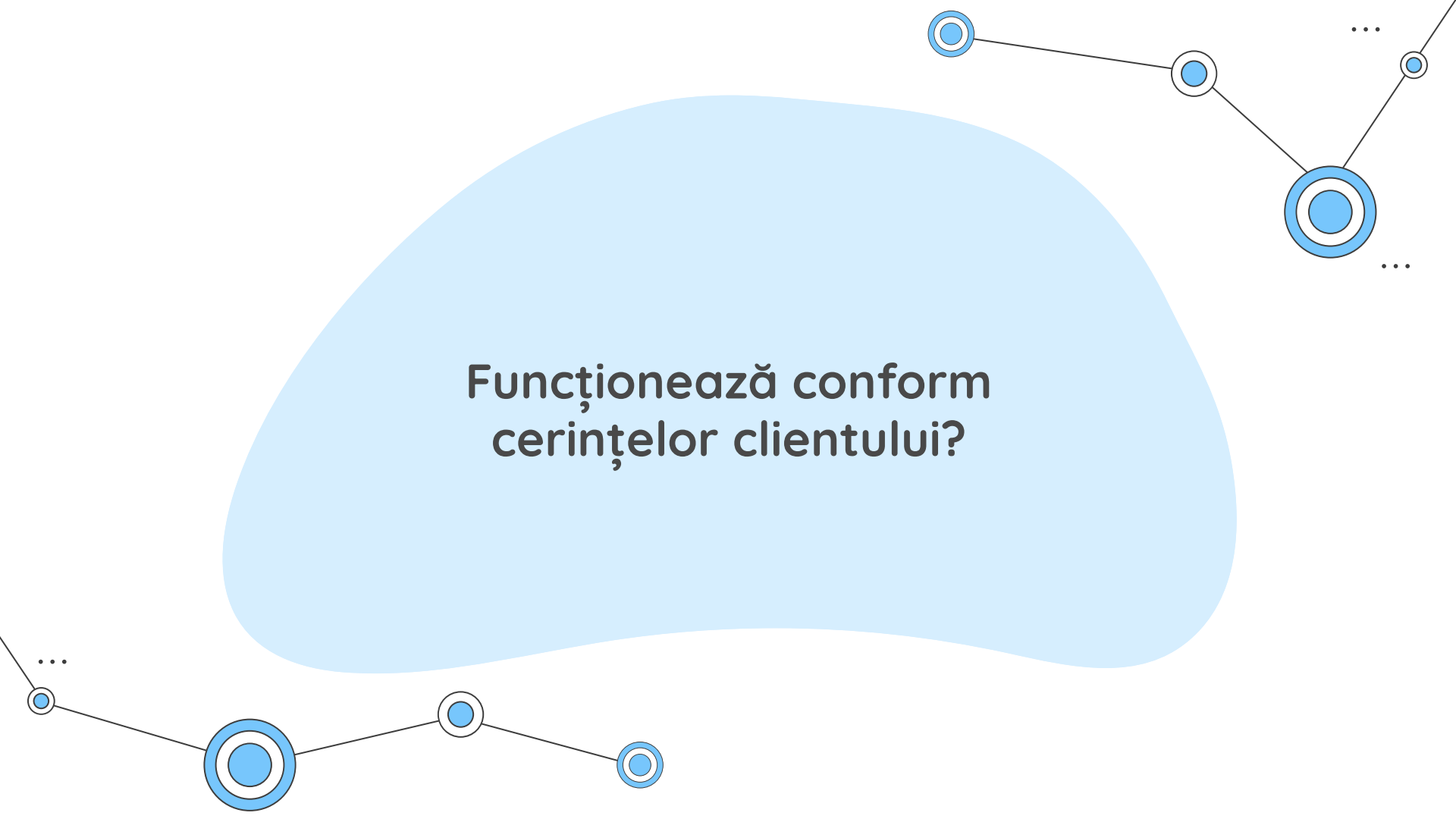
...

# Exemple de testare functionala

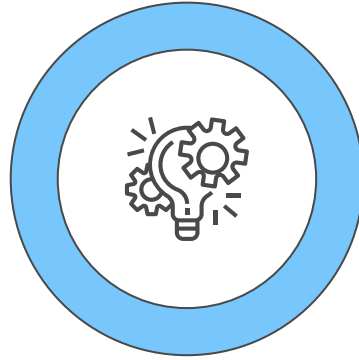
Regression testing - execută din nou teste pentru a se asigura că software-ul dezvoltat și testat anterior funcționează în continuare după o schimbare.

Usability testing - este o tehnică utilizată în proiectarea interacțiunii centrate pe utilizator pentru a evalua un produs prin testarea acestuia pe utilizatori.

...



Funcționează conform  
cerințelor clientului?



# Testarea non-funcționala

Testare efectuată pentru a evalua dacă o componentă sau  
un sistem respectă cerințele nefuncționale.  
- ISTQB 2018 Curriculum

...




# Exemple de testare non-functionala

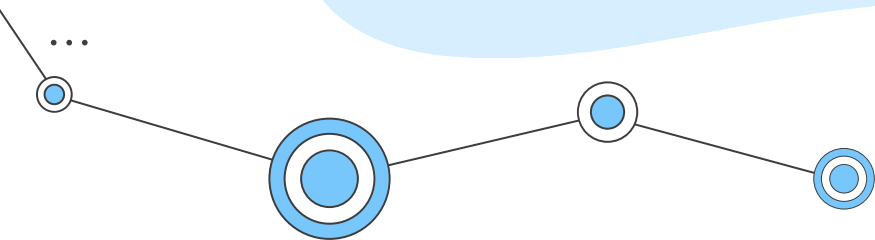
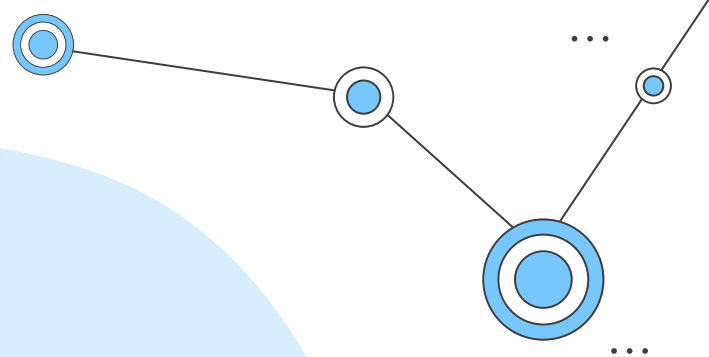
Load testing - este procesul prin care se pune presiune asupra unui sistem și se măsoară răspunsul acestuia.

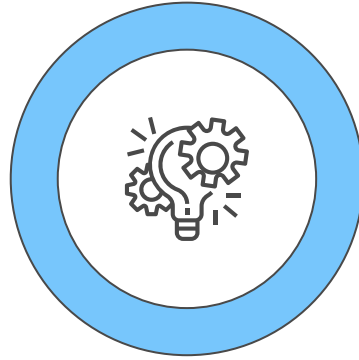
Scalability testing - este testarea unei aplicații software pentru a măsura capacitatea acesteia de a se extinde sau de a extinde în funcție de capacitatea sa nefuncțională.

...



Cât de bine  
funcționează?



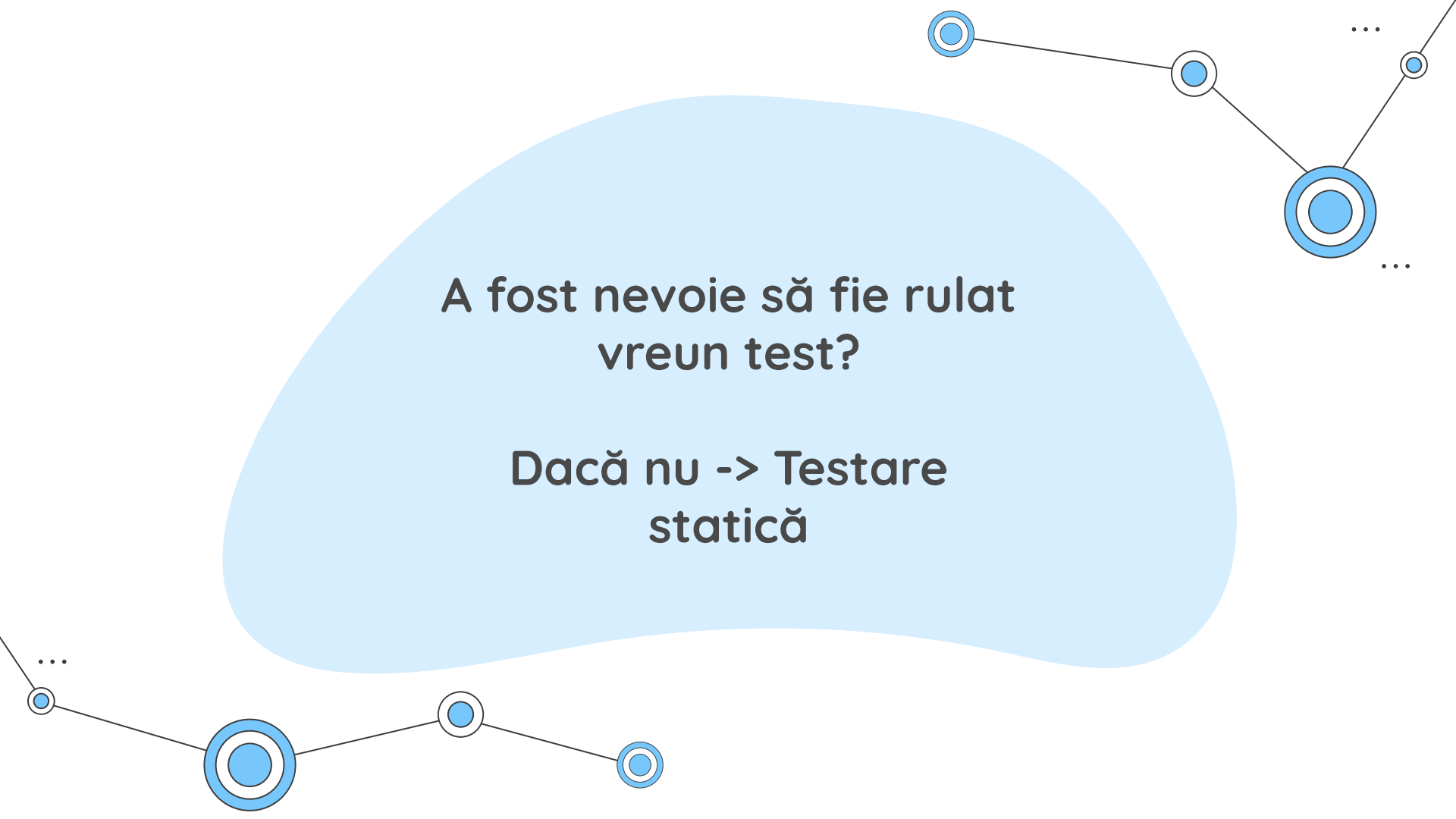


# Testare statica

Testare care nu implică executarea unui element de testare.

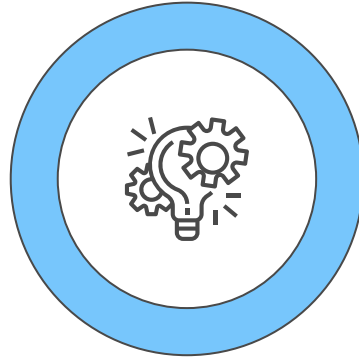
- ISTQB 2018 Curriculum

...



A fost nevoie să fie rulat  
vreun test?

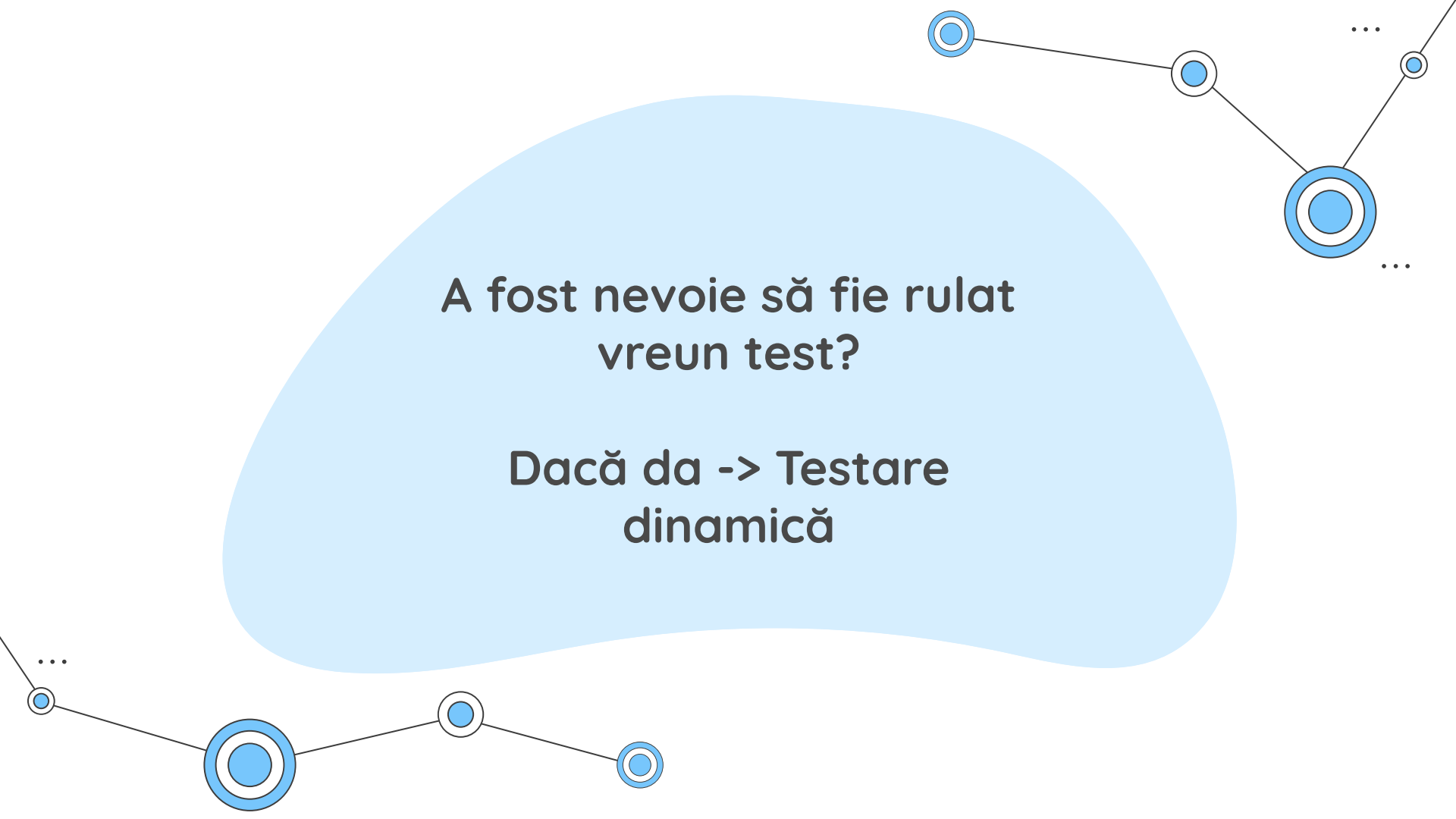
Dacă nu -> Testare  
statică



# Testarea dinamica

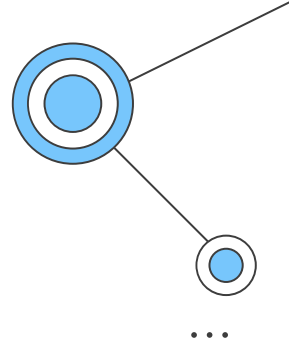
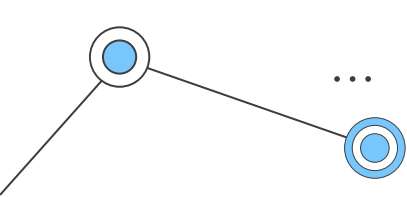
Testare care presupune executarea elementului de testare.  
- ISTQB 2018 Curriculum

...



A fost nevoie să fie rulat  
vreun test?

Dacă da -> Testare  
dinamică



01

### Testare functionala

Verifica dacă produsul software funcționează

02

### Testare non-functionala

Verifica cât de bine funcționează produsul

03

### Testare statica

Nu se creaza, ruleaza sau executa teste

04

### Testare dinamica

Se creaza, ruleaza sau executa teste

# 02

## Tehnici de testare





## Testare Black-box

Nu ne interesează codul

## Testare White-box

Ne interesează codul

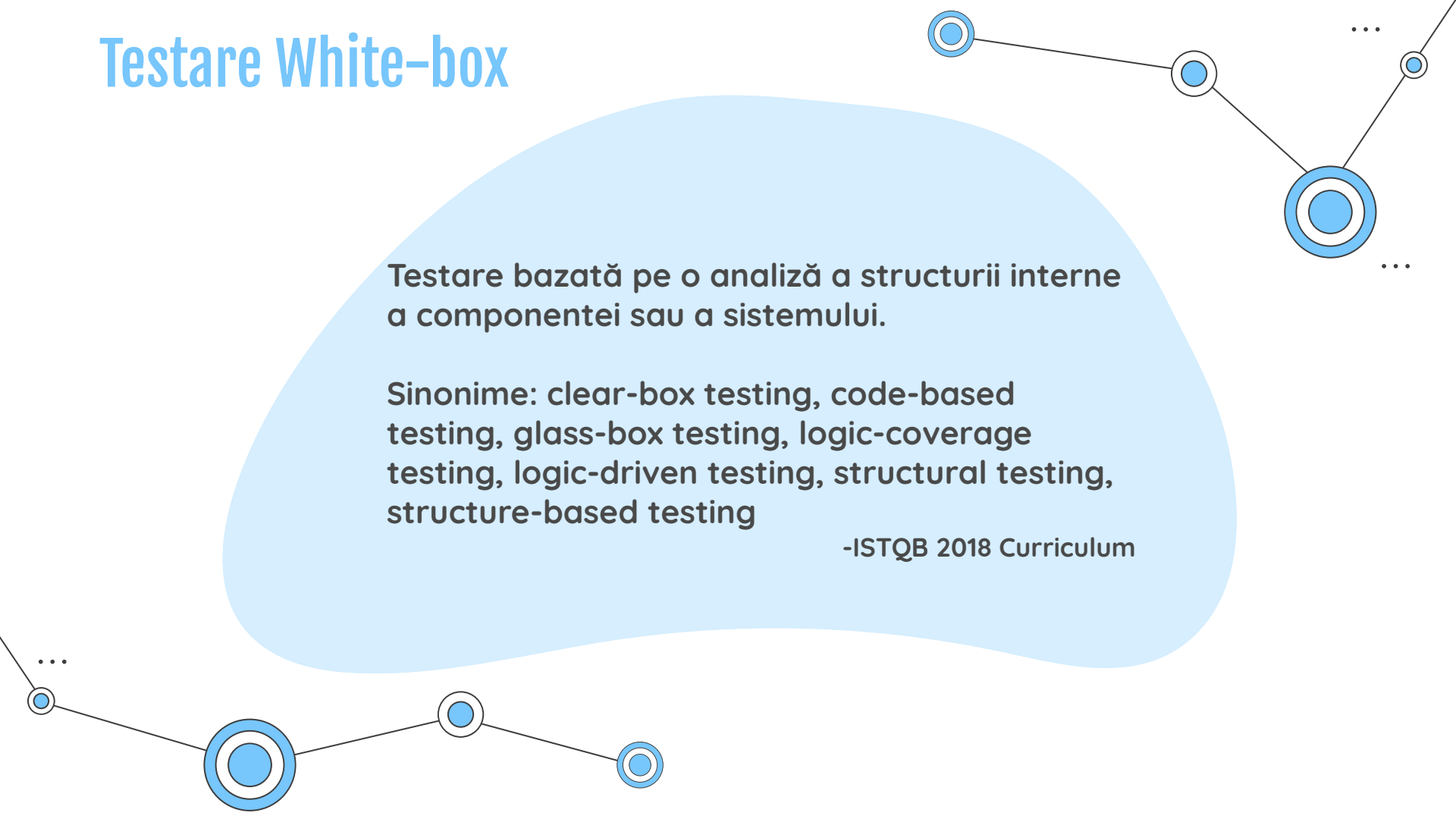


# Testare White-box

Testare bazată pe o analiză a structurii interne a componentei sau a sistemului.

Sinonime: clear-box testing, code-based testing, glass-box testing, logic-coverage testing, logic-driven testing, structural testing, structure-based testing

-ISTQB 2018 Curriculum



# Testare White-box

Input

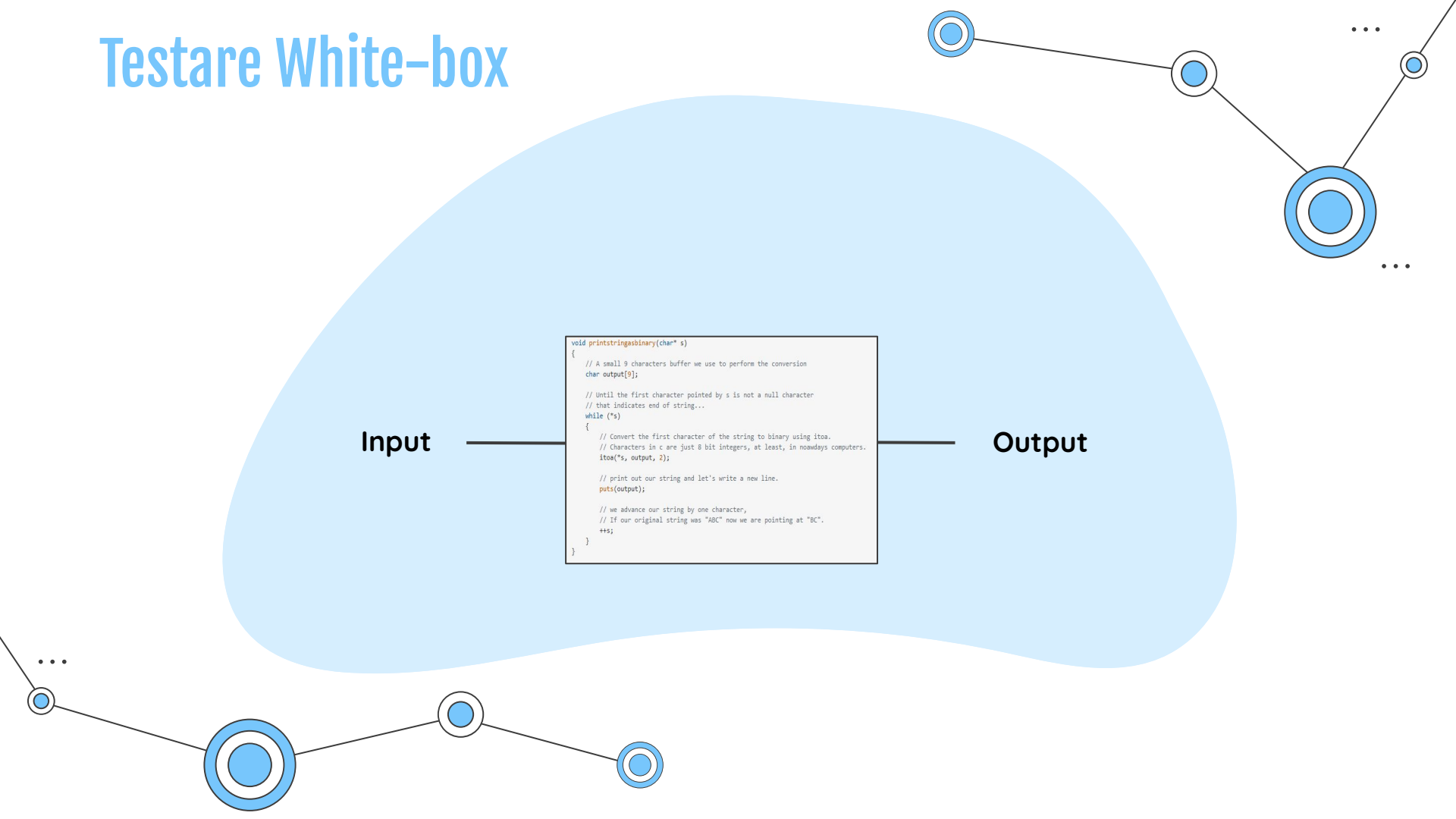
```
void printstringasbinary(char* s)
{
    // A small 9 characters buffer we use to perform the conversion
    char output[9];

    // Until the first character pointed by s is not a null character
    // that indicates end of string...
    while (*s)
    {
        // Convert the first character of the string to binary using itoa.
        // Characters in c are just 8 bit integers, at least, in nowadays computers.
        itoa(*s, output, 2);

        // print out our string and let's write a new line.
        puts(output);

        // we advance our string by one character,
        // If our original string was "ABC" now we are pointing at "BC".
        ++s;
    }
}
```

Output



# Testare White-box



Statement  
coverage



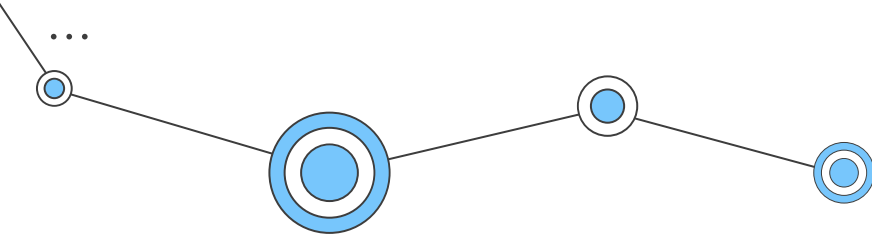
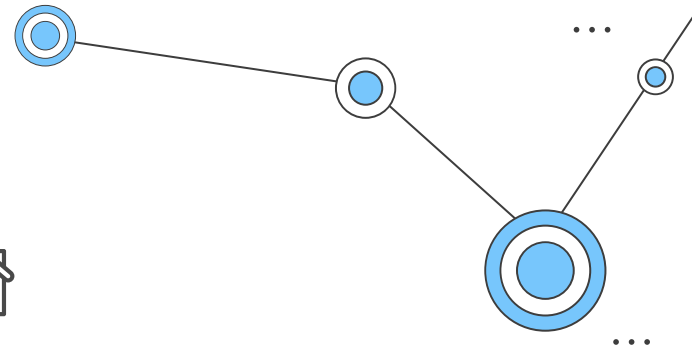
Condition coverage



Branch coverage



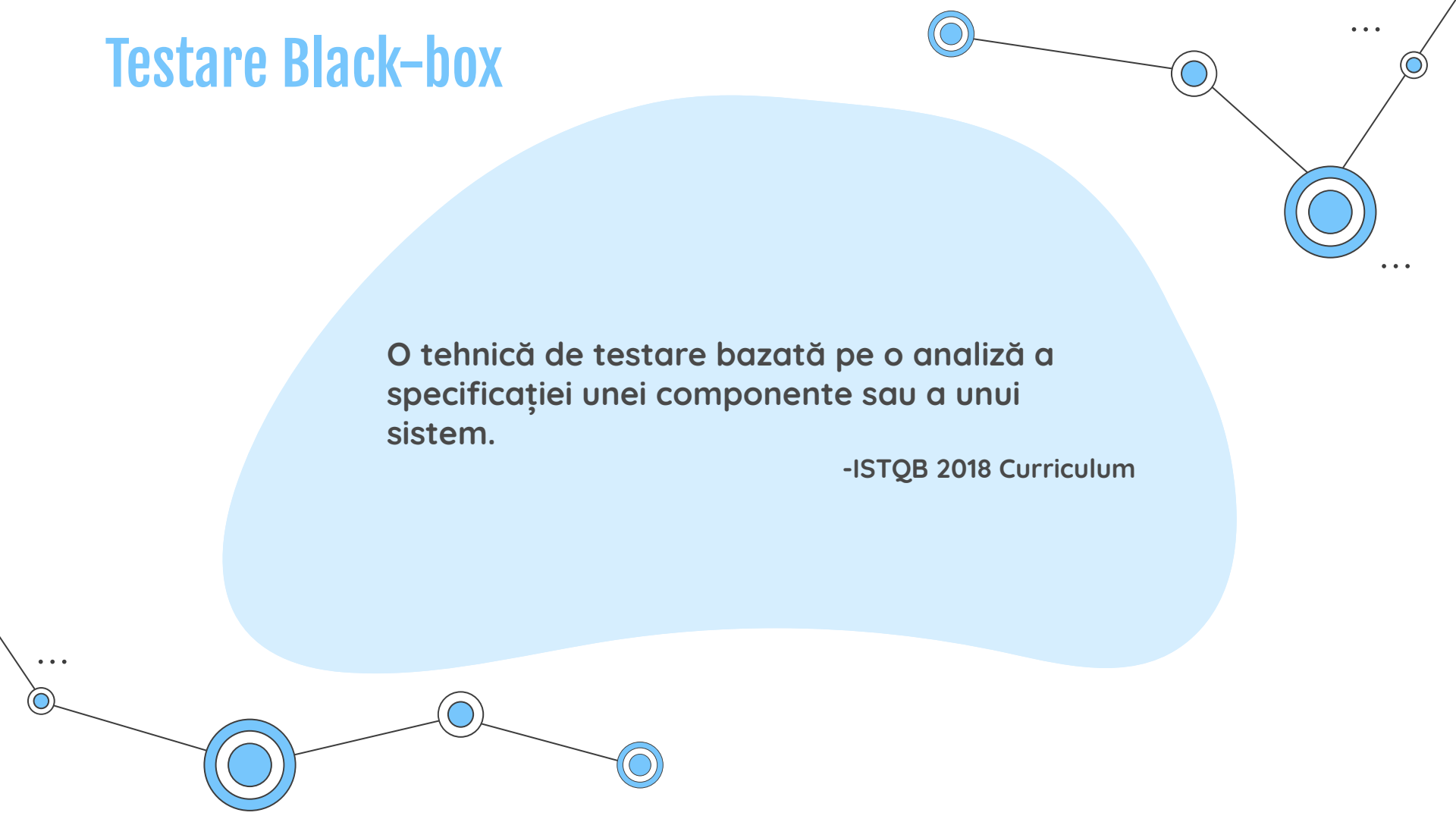
Path coverage



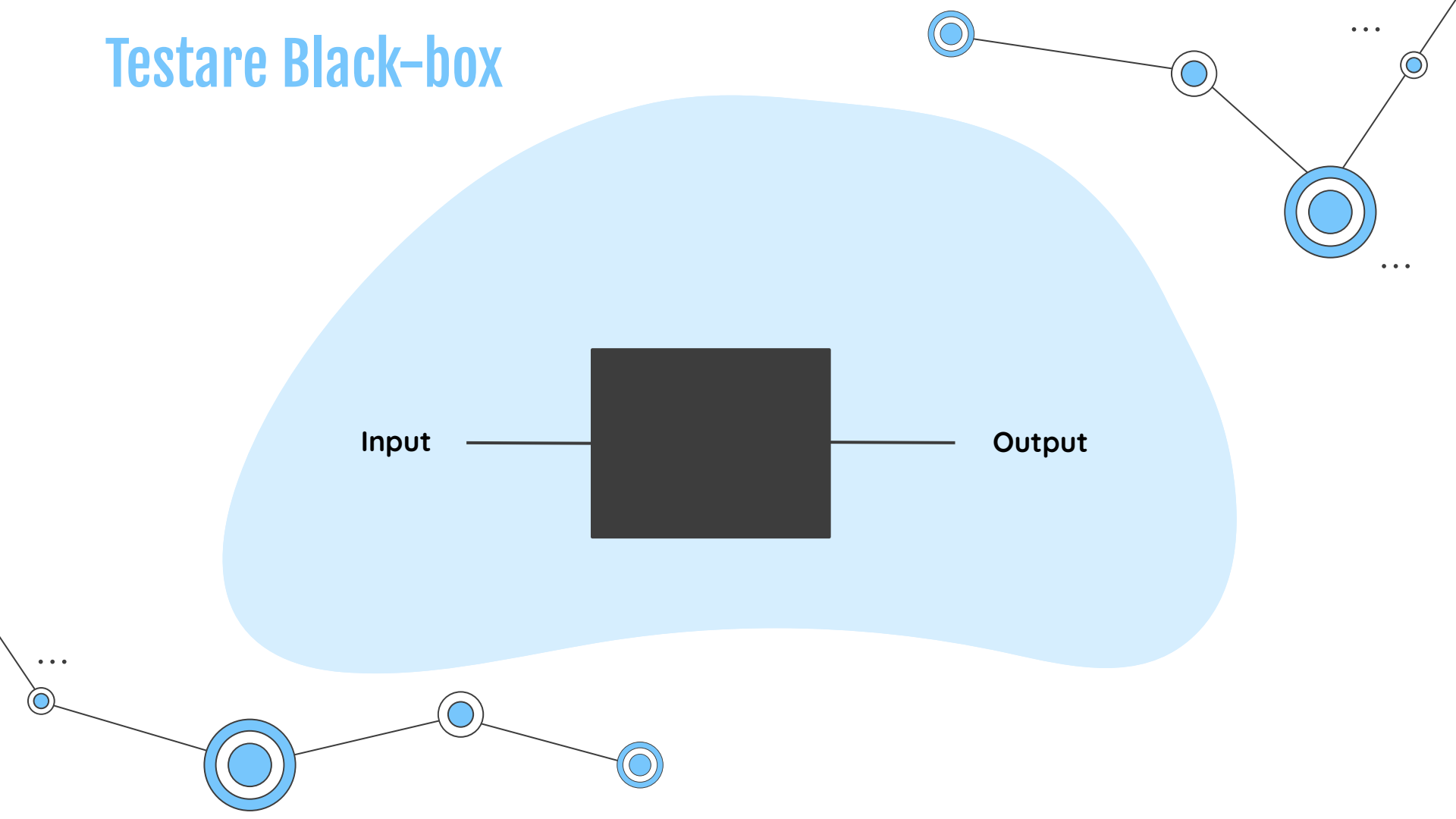
# Testare Black-box

O tehnică de testare bazată pe o analiză a specificației unei componente sau a unui sistem.

-ISTQB 2018 Curriculum



# Testare Black-box



# Testare Black-box



Use Case  
Based



Equivalence  
Class  
Partitioning



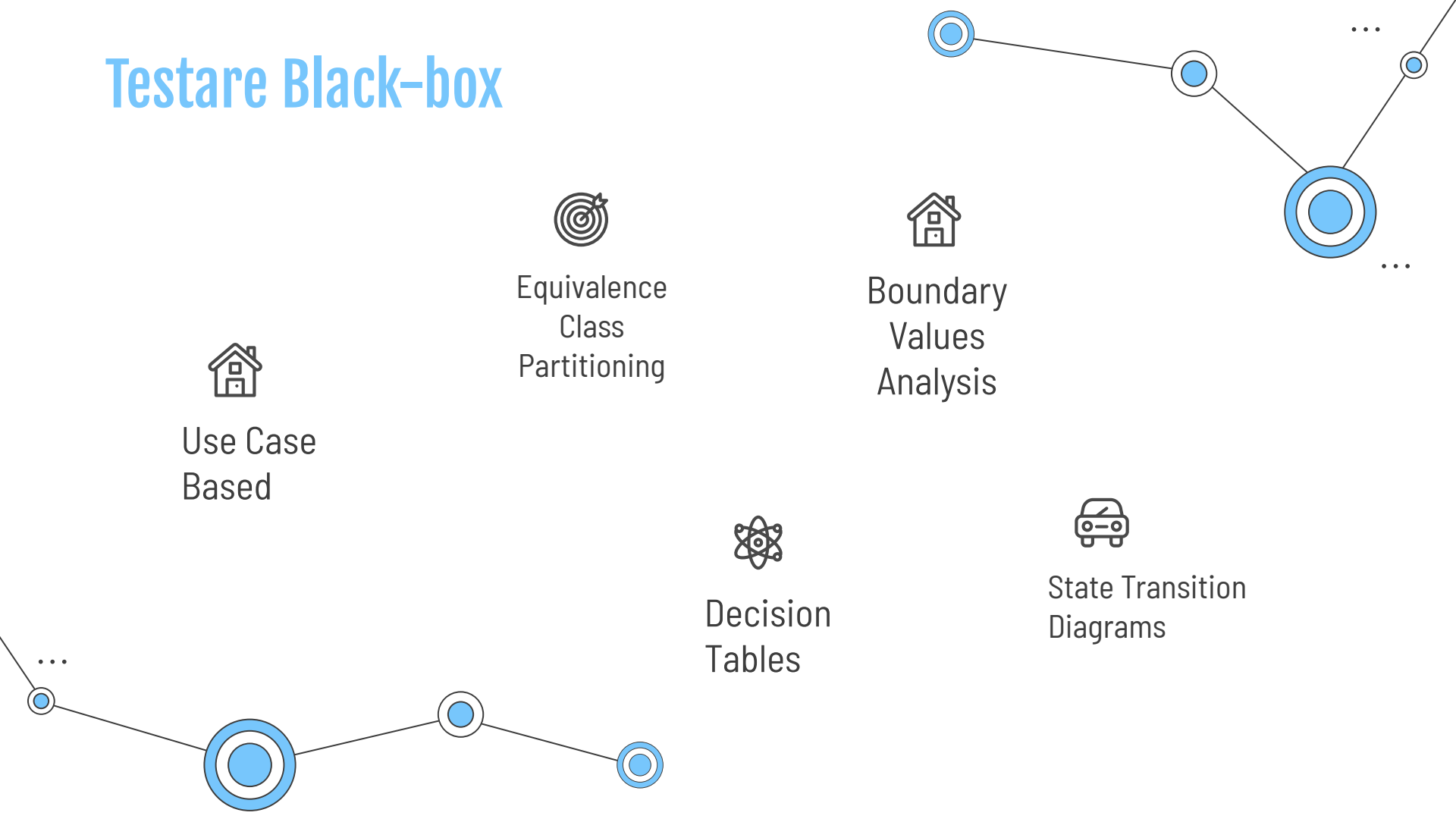
Boundary  
Values  
Analysis



Decision  
Tables



State Transition  
Diagrams



# Alte tehnici de testare



## Experience based testing

Testare bazată pe experiența, cunoștințele și intuiția testatorului.



## Exploratory testing

O abordare a testării prin care testerii proiectează și execută în mod dinamic teste pe baza cunoștințelor lor.



## Monkey testing

Este un tip de testare software în care un produs este testat folosind date de intrare aleatorii cu unicul scop de a încerca și de a strica sistemul.



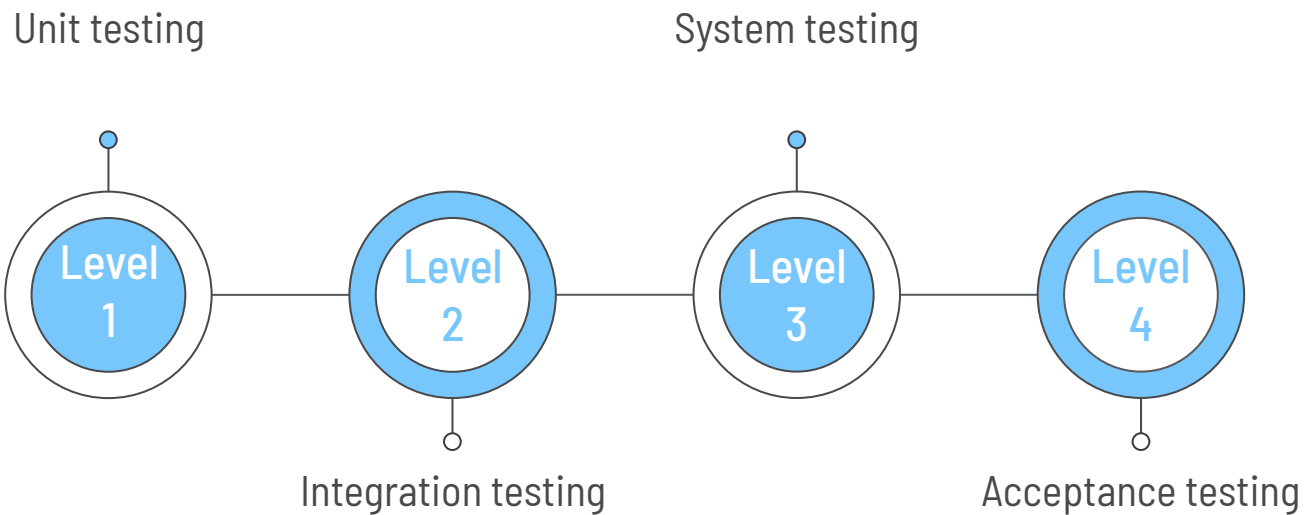


# 03

## Nivele de testare



# Our Process

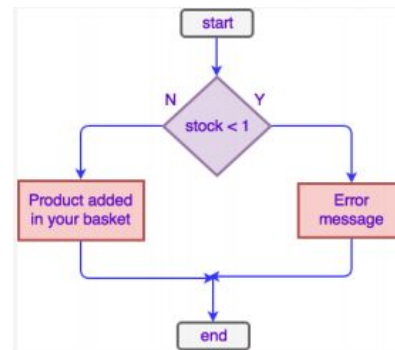


# Unit testing

Un nivel de testare care se concentrează pe componente hardware sau software individuale.

-ISTQB 2018 Curriculum

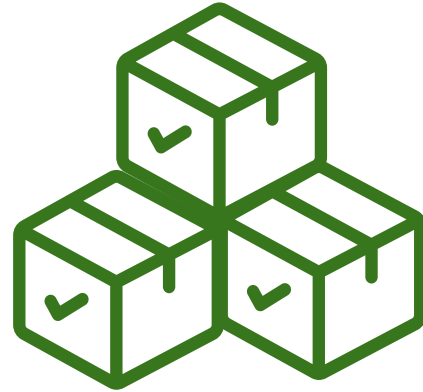
```
public void main(String[] args) {  
    int Stock = 0;  
  
    if (Stock < 1) {  
        System.out.println("Error message");  
    } else {  
        System.out.println("Product added to basket");  
    }  
}
```



# Integration testing

Un nivel de testare care se concentrează pe interacțiunile  
dintre componente sau sisteme.

-ISTQB 2018 Curriculum

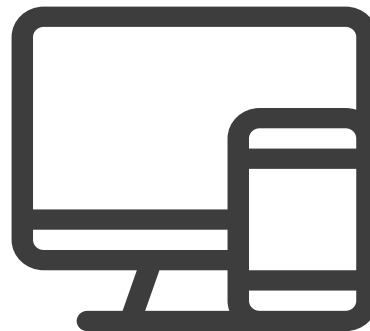
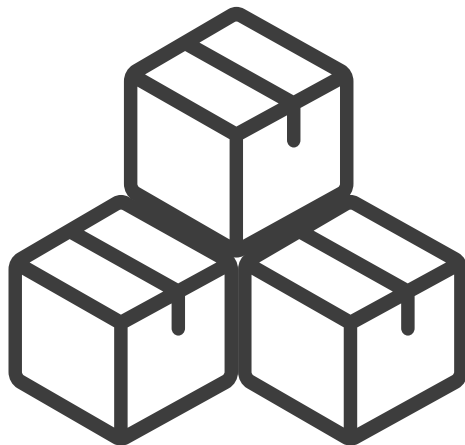


...

# System testing

Un nivel de testare care se concentrează pe verificarea faptului că un sistem în ansamblu îndeplinește cerințele specificate.

-ISTQB 2018 Curriculum

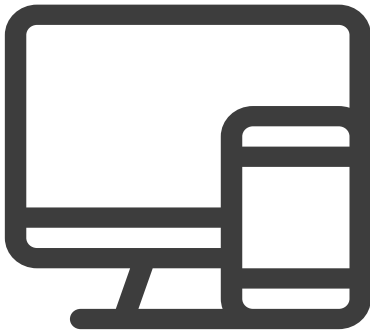


...

# Acceptance testing

Un nivel de testare care se concentrează pe determinarea  
dacă să accepte sistemul.

-ISTQB 2018 Curriculum



...

# Acceptance testing



...

# 04

## Ciclul de dezvoltare software



# Agile Manifesto

O declarație despre valorile care stau la baza dezvoltării software Agile. Valorile sunt: indivizi și interacțiuni peste procese, software de lucru peste documentație cuprinzătoare, colaborare cu clienții peste negocierea contractului, răspuns la schimbare în urma unui plan.

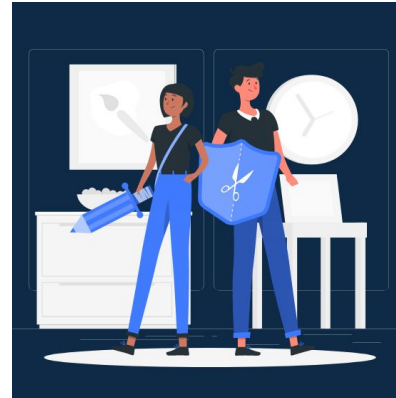
-ISTQB 2018 Curriculum



...

# Roluri

**Scrum** - Product owner, scrum master, development team



**Kanban** - Nu are roluri bine definite

# Metrici

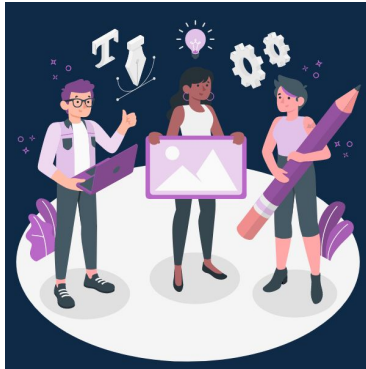
**Scrum** - Velocitate



**Kanban** - Lead time, cycle time, WIP

# Filozofia referitoare la schimbări

**Scrum** - Echipele nu trebuie să facă schimbări în timpul sprintului.

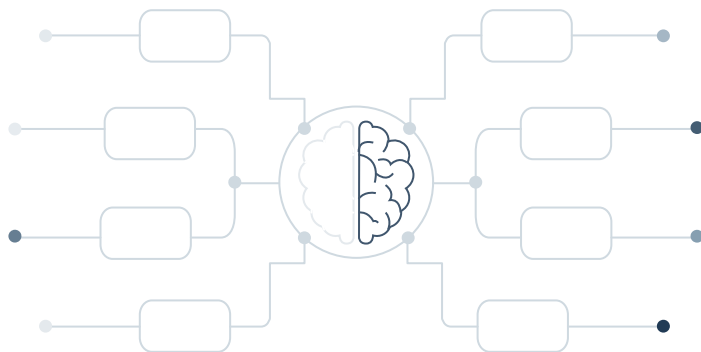


**Kanban** - Schimbarea poate avea loc oricând

# Ritm

Scrum - Sprinturi regulate cu  
lungime fixă (de exemplu, 2  
săptămâni)

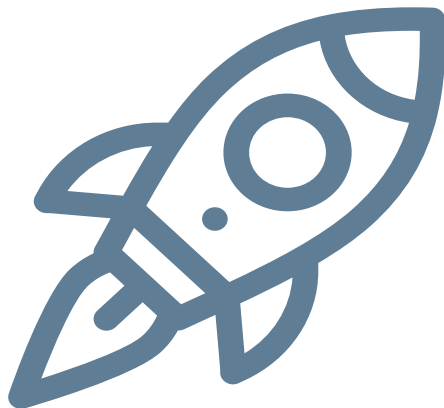
Kanban - Flux continuu



# Metodologia de lansare

**Scrum** - La final de sprint

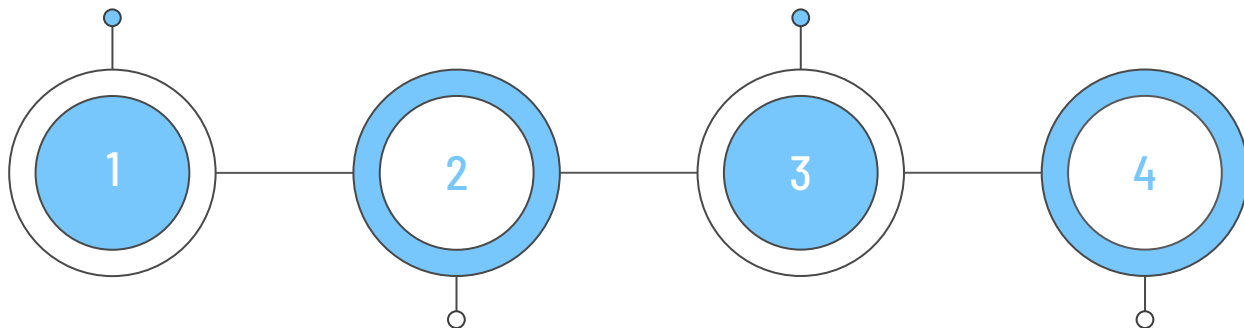
**Kanban** - Livrare continuă



# Ce este un sprint?

Se hotaraste ce task-uri  
sunt luate in sprint

Se efectuează testarea



Echipa de dezvoltare  
implementează

Se prezinta clientului

# Sprint planning

Sprint Planning inițiază Sprintul prin stabilirea lucrărilor care trebuie efectuate pentru Sprint. Acest plan rezultat este creat de munca de colaborare a întregii echipe Scrum.





# Sprint planning



Product Ownerul se asigură că participanții sunt pregătiți să discute despre cele mai importante articole din Product Backlog și despre modul în care acestea se potrivesc cu obiectivul de produs.

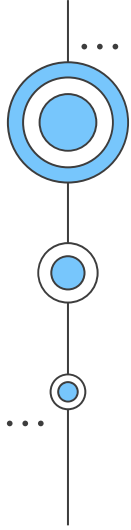
Echipa Scrum poate invita și alte persoane să participe la Sprint Planning pentru a oferi sfaturi.

# Scrum poker

Planning poker, numit și Scrum poker, este o tehnică de estimare bazată pe consens, gamificată, folosită în principal pentru estimări în principiile Agile. În planning poker, membrii grupului fac estimări jucând cărți numerotate cu fața în jos la masă, în loc să le rostească cu voce tare.

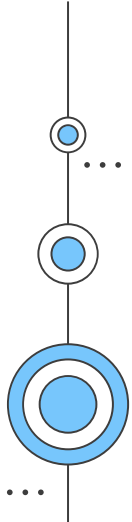
Cărțile sunt dezvăluite, iar estimările sunt apoi discutate.

Ascunzând cifrele în acest fel, grupul poate evita părtinirea cognitivă a ancorării, unde primul număr rostit cu voce tare creează un precedent pentru estimările ulterioare.



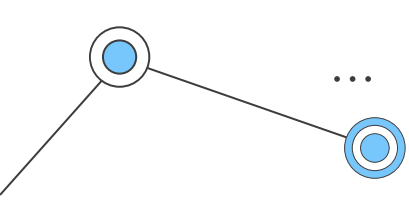
# Scrum poker exercise

<https://www.scrumpoker-online.org/en/room/190567/scrumpoker>

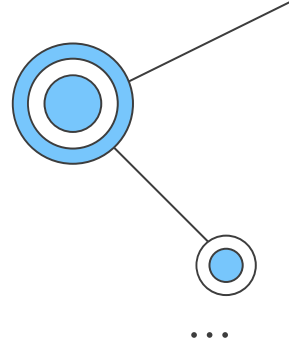


# 05

## Cele 7 principii de testare

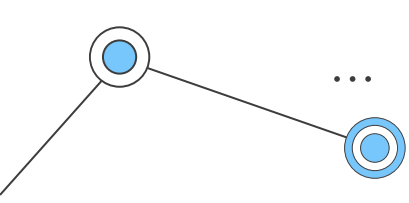


# Cele 7 principii de testare

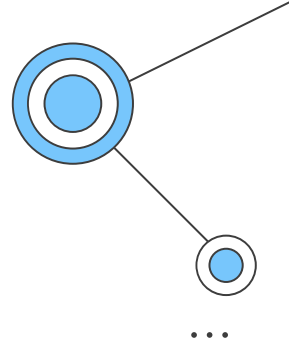


01

Testarea exhaustivă nu este  
posibilă

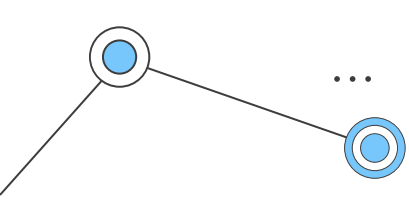


# Cele 7 principii de testare

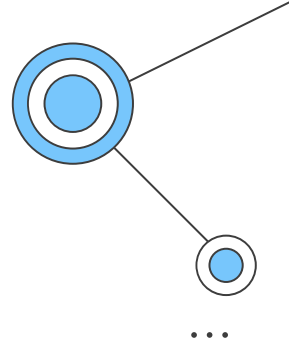


## 02

Absence-of-error fallacy

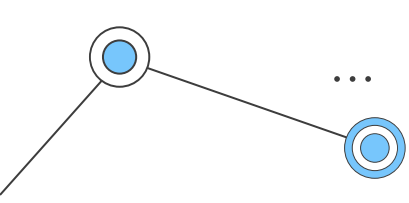


# Cele 7 principii de testare

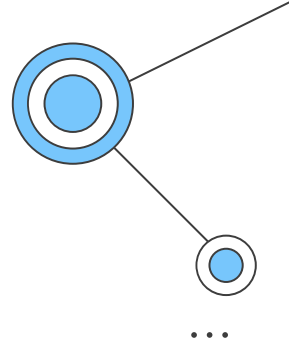


03

Testarea este  
dependentă de context



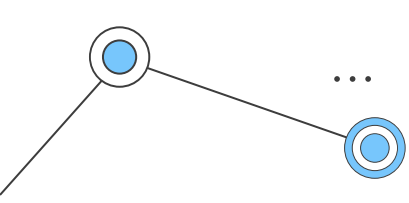
# Cele 7 principii de testare



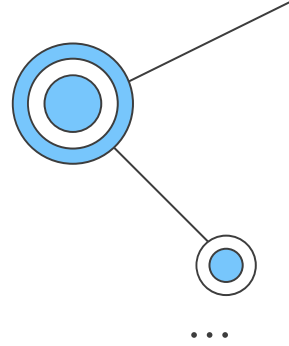
## 04

### Early testing



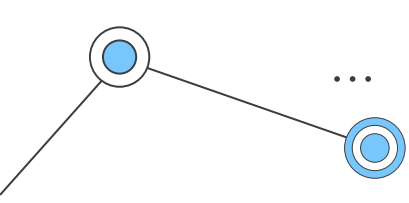


# Cele 7 principii de testare

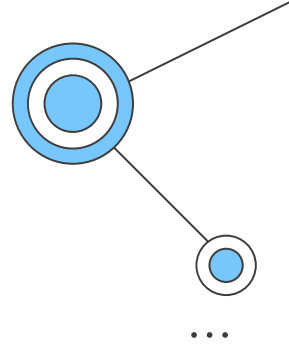


05

Paradoxul pesticidului

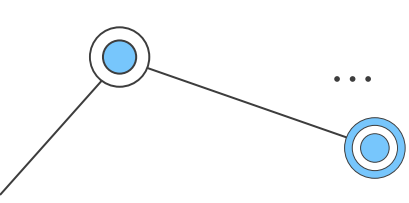


# Cele 7 principii de testare

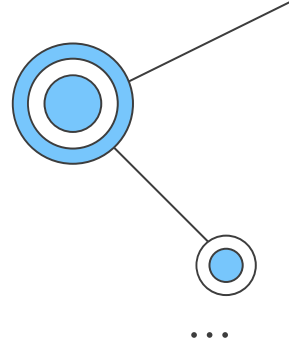


06

Defect clustering /  
Regula 80-20



# Cele 7 principii de testare



07

Testarea arată prezența  
defectelor, nu și absența lor