

Curs “Manual QA”



Codecademy
start your IT career today

Lecția 5

1. Ce înseamnă “Test design techniques”?
2. Categoriile “Test design techniques”: static & dynamic
3. Dynamic Techniques:
 - Structure – based
 - Experience – based
 - Specification – based
4. Aplicații de Team management: ServiceNow & Jira
5. Requirements tools

1. Ce înseamnă “Test design techniques”?

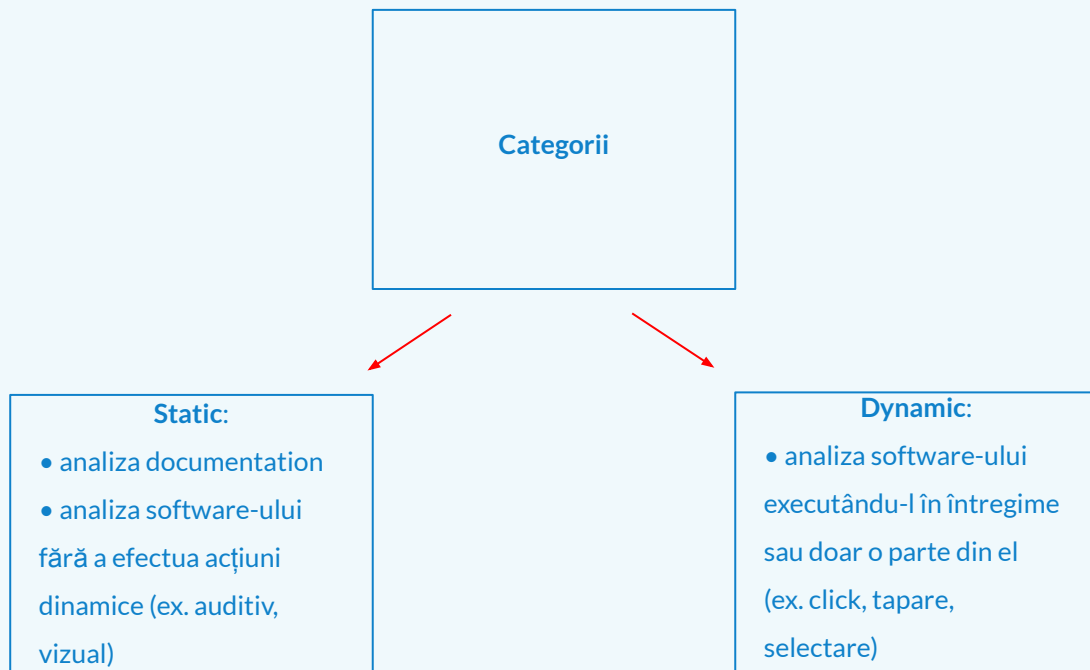
Test design techniques - tehnicile de creare a testelor necesare pentru verificarea unui produs software de către specialistul QA.

Aceste tehnici au fost create datorită faptului că testarea tuturor posibilităților existente este imposibilă și suntem obligați în calitate de specialiști QA să reducem testele efectuate.

Având scopul de a reduce testele efectuate, aceste tehnici ajută specialistul QA să-și răspundă la următoarele întrebări:

- ce funcționalități urmează să testeze
- ce funcționalități urmează să ignore și să nu testeze
- ce valori urmează să utilizeze (ex. username: denis90 parola: 984392)
- ce valori urmează să ignore și să nu utilizeze
- ce rezultat este relevant să obțin de la testele efectuate
- ce rezultat nu este relevant să obțin de la testele efectuate

2. Test design techniques



3. Dynamic techniques

Structure – based (WBT/GBT)

1. Statement (codul este executat)
2. Decision (este testat codul și rezultatul = adevărat sau fals)
3. Condition (ex. dacă este minor)
4. Multiple condition

Experience – based (BBT)

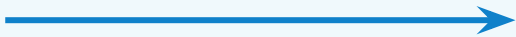
1. Exploratory
2. Error Guessing

Specification – based (BBT)

1. Equivalence partitioning
2. Boundary Values
3. Decision Tables

Daniela:

Dă un exemplu în care ai aplica tehnica statică de creare a testelor?



Elena:

Care sunt tehnicile dinamice de creare a testelor?



Experience – based

Exploratory - tehnică ce presupune analiza produsului cu scopul de a explora și cunoaște funcționalitățile produsului rezultând din experiența anterioară a specialistului QA și a crea teste pe baza acestei informații.

Error Guessing - tehnică ce presupune analiza produsului cu scopul de a căuta potențiale bug-uri rezultând din experiența anterioară a specialistului QA (a. cunoaște deja părțile slabe ale produsului b. (sau) unde anterior au existat erori).

Specification – based

Equivalence partitioning - este utilizată pentru a testa o valoare din fiecare interval (valid sau invalid, pozitiv sau negativ).

Speță:

“Permisul de conducere poate fi obținut doar de către persoanele cu vârsta cuprinsă între 18 și 56 de ani”

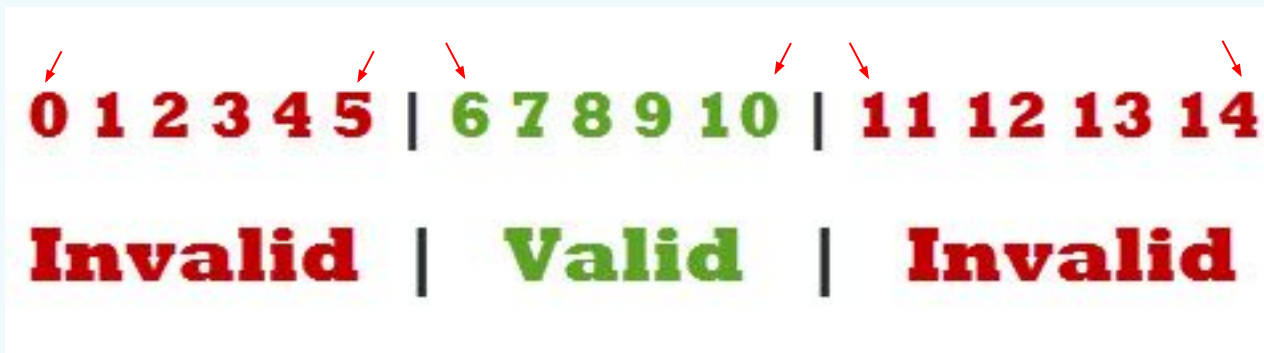
AGE: <input type="text"/> (accepts 18 to 56)		
EQUIVALANCE PARTITIONING		
INVALID	VALID	INVALID
≤ 17	18-56	≥ 57

{-10, 9, 30, 80}

Boundary Values Analysis - sunt analizate pentru testare valorile extreme din fiecare interval (valid sau invalid, pozitiv sau negativ).

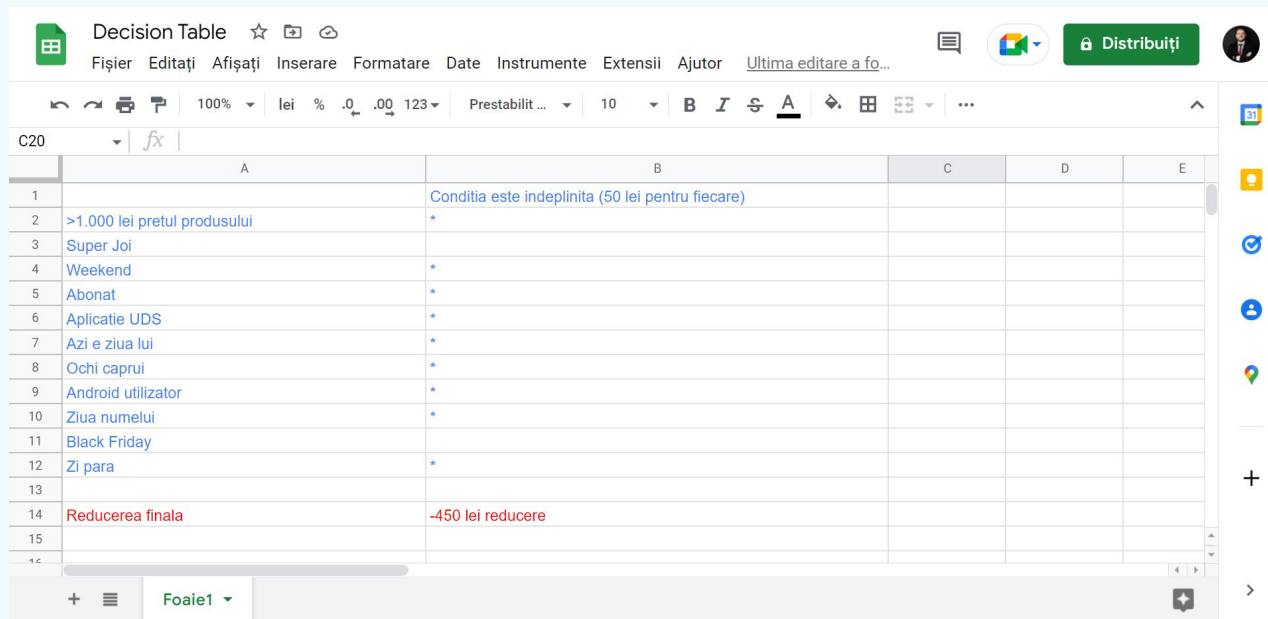
Speță:

“Călătoresc gratuit în transportul public doar copii cu vârsta cuprinsă între 6 și 10 ani. Persoanele de la 15 deja nu au voie să călătorească cu transportul public.”



{-1, 0 și 5, 6 și 10, 11 și 14, 15}

Decision Table - sunt analizate pentru testare diverse combinații de valori și condiții, inclusiv rezultatele lor cumulative.



	A	B	C	D	E
1		Conditia este indeplinita (50 lei pentru fiecare)			
2	>1.000 lei pretul produsului	*			
3	Super Joi				
4	Weekend	*			
5	Abonat	*			
6	Aplicatie UDS	*			
7	Azi e ziua lui	*			
8	Ochi caprui	*			
9	Android utilizator	*			
10	Ziua numelui	*			
11	Black Friday				
12	Zi para	*			
13					
14	Reducerea finala	-450 lei reducere			
15					

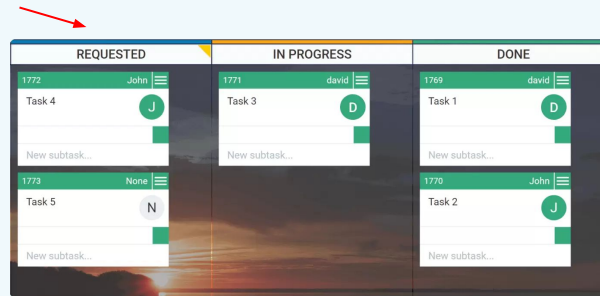
4. Aplicații de Team management

Sunt utilizate pentru:

- ❖ recepționarea task-urilor de către angajați
- ❖ comunicarea între angajați

Cele mai importante și stabile (bugs free) aplicații sunt:

- ServiceNow (utilizat mai des pentru Scrum)
- Jira (utilizat și pentru Kanban: nu există timeline, doar task-uri cerute & în desfășurare & lansate)




5. Requirements tools

Caracteristici specifice:

- ❑ platforme online ce pot fi accesate de pe orice dispozitiv
- ❑ la ele au acces doar angajații companiei (securizate / e nevoie de permisiune)

Cele mai importante instrumente sunt:

- Notion - cerințe formate la persoana a 3-a (ex. Student can buy a book from the university's website)
- Jira Confluence - cerințe formate la persoana a 1-a (în format de User Story: "As a student, I can buy a book from the university's website")



	User Story or Requirement	Importance	Release	Jira Issue	Notes
1	As an author, I need to manage the Kindle books in an online bookshelf	HIGH	Release 1	<input checked="" type="checkbox"/> BBT-1: Create a Bookshelf for KDP Editor IN PROGRESS	<input checked="" type="checkbox"/> Bookshelf Editor
2	As an author, I can add a book to the bookshelf	MEDIUM	Release 1	<input checked="" type="checkbox"/> BBT-5: Add a book to the bookshelf IN PROGRESS	How many rows per bookshelf?
3	As an author, I can delete a book from the bookshelf	MEDIUM	Release 2	<input checked="" type="checkbox"/> BBT-6: Delete a book from the bookshelf TO DO	Prompt to delete?
4	As an author, I can update the HTML code for the online book description	LOW	Release 2	<input checked="" type="checkbox"/> BBT-7: Update KDP HTML from the bookshelf TO DO	Need to store HTML in database
5					