

Răspuns întrebări studiu in echipa 5

1. **Ce înseamnă BlackBox testing?** *R: Testare fara acces la cod*
2. **La ce ne folosește tehnica de testare equivalence partitioning?** *R: La a avea o acoperire cat mai mare a aplicatiei cu un numar cat mai mic de teste cu putinta*
3. **Care este tehnica de testare care se aplica INTOTDEAUNA atunci cand se aplica equivalence partitioning?** *R: Equivalence partitioning este complementara cu Boundary Value Analysis si se vor aplica mereu impreuna*
4. **Cand ne este de folos tehnica de testare State Transition Testing?** *R: Atunci cand aplicatia trece dintr-o stare in alta in urma unui eveniment si vrem sa verificam daca acea tranzitie se face in mod corect.*
5. **Cand folosim tehnica de testare Decision Testing?** *R: Atunci cand avem mai multe scenarii care se desfasoara in functie de mai multe criterii. Spre ex: logarea in aplicatie se face doar daca atat userul este corect (criteriul 1) cat si parola este corecta (criteriul 2)*
6. **Care e diferenta intre test condition si precondition?** *R: Test condition inseamna ce testam, precondition inseamna ce criterii trebuie sa indeplinim ca sa putem sa testam*
7. **Cate test case-uri trebuie sa scriem pentru o aplicatie?** *R: Suficiente cat sa avem o acoperire cat mai mare a aplicatiei folosindu-ne de toate tipurile si tehnicile de testare cunoscute. Nu exista un numar standard*
8. **De ce trebuie sa fie conditiile de testare cat mai clare pentru testul pe care il scriem?** *R: Pentru ca ele se vor pune in titlurile testelor, iar atunci cand avem foarte multe teste si ne intereseaza unul singur, ar fi foarte time consuming sa deschidem fiecare test in parte sa ne dam seama ce contine si daca e cel care ne intereseaza*
9. **Ce sunt conditiile de testare?** *R: Sunt functionalitati specifice pe care le putem*

testa pentru a evalua comportamentul aplicatiei

10. In care etapa a procesului de testare se scriu condițiile de testare / test

case-urile? R: Condițiile de testare se scriu in etapa de analiza, iar cazurile de testare se scriu in etapa de design

11. Ce inseamna SDLC? R: Este prescurtarea de la Software Development Lifecycle si este procesul de conceptualizare, analiza, creare, testare si lansare a produsului

12. Care sunt cele doua cele mai cunoscute metodologii de SDLC? R: Agile si Waterfall

13. Care e principiila diferenta intre Agile si Waterfall? R: Agile este mai flexibil si permite modificari si in timpul procesului de dezvoltare, in timp ce waterfall este mai rigid, iar orice functionalitate ce nu a fost planificata la inceputul proiectului trebuie sa fie tratata intr-un proiect separat (change request)

14. Care dintre cele doua metodologii este mai buna? R: Nu exista una care sa fie perfecta in absolut orice situatie. Metodologia, in mod ideal, ar trebui sa fie aleasa in functie de proiectul care este implementat

15. Puteti sa dati un exemplu de situatie in care waterfall ar fi mai bun decat Agile?

R: Intr-un proiect cu deadline sau buget fix, spre ex un proiect guvernamental - GDPR

16. Care e diferenta intre un model iterativ si un model incremental? R: Intr-un model iterativ, se merge cu construirea partiala a intregii aplicatii, dupa care se continua cu completarea si imbunatatirea functionalitatilor, in timp ce la un model incremental fiecare modul / functionalitate este dezvoltata in intregime inainte de a se merge mai departe la urmatorul modul / functionalitate

17. Cum decidem daca sa folosim Agile sau Waterfall? R: In functie de specificul proiectului. Daca e un proiect cu buget / deadline fix atunci recomandat ar fi sa mergem pe Waterfall. In schimb, daca ne asteptam cu modificari frecvente din partea clientului si nu stim exact foarte bine de la bun inceput cum vrem sa arate produsul, atunci putem sa mergem pe Agile.

Răspuns întrebări studiu in echipa 7

1. **Ce inseamna Whitebox testing?** *R: Este o tehnica de testare dinamica prin intermediul careia se valideaza codul*
2. **Care e diferenta intre statement coverage si decision coverage?** *R: Statement coverage ne ajuta sa cream teste care sa acopere toate instructiunile din program, in timp ce decision coverage ne ajuta sa cream teste pentru toate ramurile decizionale din program, indiferent daca au statement-uri sau nu.*
3. **Care este mai acoperitoare intre statement si decision coverage?** *R: Decision coverage*
4. **Ce inseamna complexitatea ciclomatica?** *R: Este o masura a complexitatii codului*
5. **La ce ne foloseste complexitatea ciclomatica?** *R: Este utila mai ales in procesul de dezvoltare, pentru a intelege daca un program poate fi simplificat. Obiectivul este sa facem un program cat mai simplu si mai usor de gestionat. Pentru testare poate fi uneori utila pentru a ne face o idee cu privire la complexitatea produsului si a face estimari cu privire la durata testarii unei anumite functionalitati.*

Răspuns întrebări studiu in echipa 8

1. **Ce este Jira?** *R: Este o unealta de gestiune a proiectului (project management tool) prin intermediul careia putem sa tinem sub control ceea ce vrea clientul, ceea ce dezvoltam, problemele identificate rapoarte si orice alte elemente care ne vor ajuta sa colaboram in mod corect in procesul de dezvoltare a produsului software care este dezvoltat*
2. **Ce este Zephyr?** *R: Este o unealta de gestiune a testarii (test management tool) prin intermediul careia echipa de testare poate sa isi tina sub control toate activitatile cum ar fi crearea de teste, executarea lor, raportarea statusului testarii si observarea nivelului de acoperire a cerintelor de business*
3. **La ce ne foloseste un release?** *R: Pentru a putea aduna intr-un singur loc toate functionalitatile care trebuie sa fie lansate la o anumita data pentru un anumit proiect. Este util atunci cand intr-o organizatie lucreaza mai multe echipe la diverse proiecte. Atunci cand avem mii de story-uri, taskuri, bug-uri etc in diverse statusuri (inclusiv vechi, cu statusul closed), devine greu de urmarit ce este in scopul nostru la un anumit moment in absenta unui release*
4. **De ce avem nevoie de epic-uri, story-uri, task-uri si subtaskuri?** *R: Pentru a putea descrie mai usor functionalitatile care se doreste a fi implementate. Daca am pune toate informatiile in epic-uri, de exemplu, am avea un singur loc in care sunt scrise zeci sau sute de randuri, care ar fi mai greu de citit. In acest caz, spargem in bucatile in bucati mai mici in care functionalitatile descrise sunt sparge in bucati mai mici astfel incat granularitatea (nivelul de detaliu) sa fie din ce in ce mai mult, pe masura ce trecem de la epic la story -> task -> subtask*
5. **De ce nu se adauga testele in release?** *R: Pentru ca release-ul trebuie sa contina elemente care sunt de interes pentru toata echipa, gen epic-uri, story-uri, task-uri, subtask-uri, bug-uri. Testele sunt de interes doar pentru echipa de testare, iar daca se doreste a fi transmise informatii punctuale spre exterior se pot transmite fie sub forma de rapoarte sau alte elemente depinzand de ceea ce se doreste. Avand in vedere ca testele acopera o plaja foarte mare de scenarii, inclusiv corner cases care nu au fost specificate in cerinte, este posibil ca release-ul sa devina foarte aglomerat si greu de gestionat*
6. **La ce ne foloseste un raport de executie din dashboards?** *R: Pentru a evalua la finalul procesului de testare statusul general si a intelege daca la final in continuare mai sunt teste cu statusul fail si de a intelege daca exista ceva riscuri asociate cu acestea*
7. **La ce ne foloseste un raport de executie din test metrics?** *R: Pentru a vedea felul in care a evoluat*

testarea si statusul testelor intr-o anumita perioada de timp

8. **De ce adaugam testele intr-un cycle summary?** *Din acelasi motiv din care facem release-uri, pentru ca la nivel de organizatie se vor face mii de teste, si atunci noi trebuie sa putem sa stim ce testam in scopul unui anumit proiect, astfel incat sa putem sa stim ce testam, cat testam, ce status are testarea, si sa putem sa controlam mai usor eventualele situatii neprevazute in cazul in care acestea apar*
9. **Care e un motiv pentru care bug-urile nu sunt vizibile in dreptul testelor / story-urilor in matricea de trasabilitate?** *R: Pentru ca legarea bug-urilor la teste prin linked issue nu e suficient, ele trebuie sa fie legate la teste din modulul de executie*

Răspuns întrebări studiu în echipa 10

1. **Ce este API?** *E: Este prescurtarea de la Application Programming Interface si reprezinta o colectie de protocoale si reguli de transmitere a informatiei care asigura comunicarea intre doua sisteme*
2. **Cine este responsabil cu crearea API-ului?** *R: Echipa de dezvoltare*
3. **Cine e responsabil cu validarea API-ului?** *R: Echipa de testare*
4. **Care e documentul care stă la baza testării de API (ce trebuie sa testam)?** *R: Documentatia de API care descrie cum trebuie sa functioneze*
5. **Care sunt principalele metode HTTP?** *R: GET, POST, PUT, PATCH, DELETE*
6. **Care sunt de regula metodele HTTP care cer folosirea unui request body?** *R: POST, PUT, PATCH*
7. **Care e diferenta intre PUT, POST, PATCH?** *R: POST - Creaza resurse noi. Daca resursa deja exista returneaza eroare. PUT: Actualizeaza resurse existente in intregime. Daca resursa nu exista o creeaza. PATCH: Actualizeaza partial resurse existente.*
8. **Care e codul de raspuns returnat de metodele principale de HTTP (GET, POST, PUT, PATCH, DELETE)?** *R: GET returneaza 200. POST, PUT, PATCH returneaza 201. DELETE returneaza 204.*
9. **Care e diferenta intre codul de raspuns 401 si respectiv 403?** *R: 401 inseamna unauthorised, adica nu suntem logati in aplicatie. 403 inseamna forbidden, adica suntem logati in aplicatie dar nu avem dreptul sa accesam resursa respectiva*
10. **Ce este postman?** *R: Este o unealta prin intermediul careia putem trimite requesturi de API pentru a observa comportamentul lor, astfel incat sa evaluam corectitudinea functionarii acestora*

Răspuns întrebări studiu în echipa 11

1. **Ce este un request?** *R: Este un set de informatii care sunt transmise sub forma de mesaj catre un server pentru a cere indeplinirea unei actiuni, fie ea de creare sau actualizare resurse sau extragere informatii despre resurse existente*
2. **Care sunt componentele unui request?** *R: Endpoint (i se mai spune si url = uniform resource locator sau uri = uniform resource identifier), metode http (GET, POST, PUT, PATCH, DELETE etc), parametri (de regula utilizati pentru filtrare), request body (folosit in general pentru requesturile de creare sau actualizare de resurse)*
3. **Ce este un token?** *R: Este un cod care va reprezenta un "contract" intre client (calculatorul de pe care lucreaza utilizatorul) si server (locul unde se proceseaza de fapt cererea) si care autentifica faptul ca acel request este valid*
4. **Ce este un request URL?** *R: Este linkul care este scris pentru a defini locul in care requestul trebuie sa ajunga in server pentru a putea fi procesat (practic locul in care se afla resursa pe care vrem sa o accesam)*
5. **Ce este un endpoint?** *R: Este de regula acelasi lucru cu URL-ul.*
6. **Ce este o variabila globala si la ce foloseste?** *R: Este un loc in care se stocheaza niste informatii necesare mai multor requesturi. Este folosita atunci cand acele informatii au potentialul de a se schimba foarte des, oferindu-ne posibilitatea de a modifica intr-un singur loc si de a reproduce acea modificare in toate requesturile care fac referinta catre acea informatie*
7. **Ce este collection runner?** *R: Este o componenta a aplicatiei Postman care permite rularea in mod automat a tuturor requesturilor dintr-o colectie*
8. **Care sunt componentele raspunsului unui request?** *R: coduri de status (ex: 200, 201, 401, 403 etc), response header (contine parametri de sistem care sunt returnati in urma executarii requestului) si usually response body (care contine resursa care a fost creata / modificata / extrasa sau un mesaj de eroare.*

- 9. De unde stim ce trebuie sa validam la un request?** *R: Facem referinta la documentatia de API care ne spune ce trebuie sa contina un request (metode http, parametri, endpoint, request body, response body etc), iar noi vom valida faptul ca asteptarile definite in documentatia de API sunt intr-adevar indeplinite*
- 10. Cine e responsabil cu crearea documentatiei de API?** *R: Echipa de dezvoltare*

Răspuns întrebări studiu în echipa 12

1. **Ce sunt testele in Postman?** *R: Sunt linii de cod scrise in limbajul java script care pot valida in mod automat raspunsul returnat de un request de API*
2. **Ce putem evalua cu ele?** *R: In general sunt folosite pentru a evalua status code-ul, response body sau alte elemente din response-ul unui request de API*
3. **De ce trebuie sa scriem teste in postman? Care e utilitatea lor?** *R: Ne ajuta in efectuarea unei testari mai rapide si mai eficiente intr-un timp mai scurt*

Răspuns întrebări studiu în echipa 14

- 1. Care e diferența între o cheie primară și o cheie secundară?** *R: O cheie primară este o modalitate de a furniza o modalitate unică de identificare a fiecărui rând dintr-o tabelă a unei baze de date, în schimb ce o cheie secundară este o referință către acea cheie primară care face legătura între două tabele.*
- 2. Ce sunt funcțiile agregate?** *R: Funcțiile agregate sunt modalități prin care se pot procesa mai multe rânduri într-un singur rezultat (exemplu, salariile mai multor angajați se pot procesa într-un singur rezultat de tip medie aritmetică)*
- 3. Ce se va returna în urma următoarei instrucțiuni? De ce se întâmplă asta?**
Select department, count(employeeid) from departments d inner join employees e on d.departmentid = e.departmentid *R: Va returna eroare, deoarece selectul conține și alte coloane în afara funcției agregate, care nu au fost incluse într-o clauză group by. Astfel, sistemul nu știe pentru care departament să returneze număratoarea. Prin adăugarea unui group by anunțăm sistemul că vrem să ni se calculeze numărul de angajați pentru fiecare departament, și apoi să afișeze fiecare departament împreună cu numărul de angajați corespunzător*
- 4. Care e diferența între while și having?** *R: While se folosește pentru a face filtrări pe valorile unei coloane, iar having se folosește pentru a face filtrări pe rezultatele unei funcții agregate (de exemplu, returnează toate departamentele care au mai mult de 20 de angajați)*
- 5. Care este ordinea priorității între operatorii AND, OR, NOT?** *R: NOT are cea mai mare prioritate, urmat de AND și apoi de OR. Ordinea priorității se poate schimba în funcție de necesități folosind parantezele, la fel cum putem schimba prioritatea operatorilor de adunare / scădere / înmulțire / împărțire*
- 6. Ce este un cross join?** *R: Este o modalitate de a extrage datele din două sau mai multe tabele returnând produsul cartezian între acele două tabele, adică făcând combinații între toate rândurile din toate tabelele. În general nu este recomandat*

decat in situatii exceptionale deoarece poate sa cauzeze probleme serioase de performanta.

Răspuns întrebări studiu în echipa 15

- 1. La ce ne folosește un autoincrement?** *R: La a automatiza generarea cheii primare pentru a nu fi nevoiti sa o inseram in mod explicit*
- 2. Ce va returna instructiunea `ALTER TABLE users modify user_id auto_increment`?**
De ce? *R: Va returna eroare, deoarece instructiunea de modify va modifica parametrii existenti ai unei coloane cu cei specificati in query, iar cei nespecificati vor fi stersi. Avand in vedere ca tipul de data este o informatie obligatorie pentru o coloana, acesta nu poate fi sters, drept urmare, sistemul nu va putea executa instructiunea de actualizare*
- 3. Ce se intampla daca adaugam un rand intr-o tabela care are pe cheia secundara o valoare care nu exista pe cheia primara? De ce?** *R: Va returna eroarea Cannot add or update a child row: a foreign key constraint fails deoarece va incerca sa insereze o referinta catre o valoare care nu exista*
- 4. Ce se intampla daca incercam sa stergem un rand dintr-o tabela care are pe cheia primara valori care exista pe cheia secundara?** *R: Va returna eroarea Cannot delete or update a parent row: a foreign key constraint fails deoarece se va incerca sa se stearga o inregistrare care este referentiata intr-o alta tabela (spre ex: nu putem sa stergem clienti care au inca facturi in baza de date*
- 5. Ce se intampla daca incercam sa stergem valoarea de pe cheia secundara a unei inregistrari dintr-o tabela? De ce?** *R: Vom putea sa o stergem pentru ca in cazul asta nu exista nicio restrictie. Logica: intai adaugam pe cheia primara inainte sa adaugam orice pe cheia secundara, intai stergem de pe cheia secundara tot inainte sa stergem cheia primara*
- 6. Ce face instructiunea truncate? De ce returneaza la Action output: 0 rows affected?** *R: Pentru ca truncate este o instructiune de DDL si nu numara inregistrarile pe care le sterge*

Răspuns întrebări studiu în echipa 16

1. Care e diferența între instrucțiunile **insert into nume_tabela (coloana_1, coloana_2) values (value_1, value_2)** și **insert into nume_tabela values (value_1, value_2)**? R: La prima instrucțiune nu vrem să inserăm musai valori pe toate coloanele, și atunci specificăm în mod explicit pe care coloane vrem să le actualizăm
2. Ce înseamnă eroarea: **Cannot add or update a child row: a foreign key constraint fails (client_id, CONSTRAINT client_id fk FOREIGN KEY (client_id) REFERENCES Clients (client_id))**? R: Înseamnă că am încercat să interacționăm (să adăugăm o înregistrare într-o tabelă care are pe cheia secundară o valoare ce nu există pe cheia primară)
3. Ce înseamnă eroarea: **Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`restaurants_management`.`dish`, CONSTRAINT `dish_ibfk_1` FOREIGN KEY (`restaurant_id`) REFERENCES `restaurant` (`id`))** R: Înseamnă că am încercat să ștergem o valoare de pe cheia primară înainte să ștergem corespondenții de pe cheia secundară.
4. Care e diferența între cele două erori (când apare fiecare)? R: Prima apare când încercăm să adăugăm o valoare pe cheia secundară care nu există pe cheia primară, a doua când vrem să ștergem de pe cheia primară valori care au încă referințe pe cheia secundară

Răspuns întrebări studiu în echipa 17

1. Care dintre operatorii AND, OR si NOT au prioritate mai mare? R: NOT, apoi AND si apoi OR
2. Care e diferenta intre like "%valoare", like 'valoare%' si like '%valoare%'? R: %valoare cere o resursa de tip text care se termina cu valoarea specificata, 'valoare%' cere o resursa de tip text care incepe cu valoarea specificata iar '%valoare%' cere o resursa de tip text care contine valoarea specificata
3. De ce avem nevoie de group by? R: Pentru a specifica sistemului faptul ca vrem ca rezultatele unei functii agregate sa fie grupate in functie de anumite coloane
4. Ce inseamna eroarea urmatoare: Error Code: 1140. In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'restaurants_management.restaurant.id'; this is incompatible with sql_mode=only_full_group_by? R: Inseamna ca nu putem sa adaugam intr-un select, pe langa o functie agregata, coloane care nu sunt specificate intr-o clauza group by
5. La ce ne ajuta joinurile? R: La a extrage informatii din mai multe tabele in acelasi timp
6. La ce ne ajuta un subquery? R: La a trata rezultatul unui query ca o tabela temporara ce poate sa fie folosita, spre exemplu, in joinuri