**Release Notes**

Súhrn zmien v novej verzii aplikácie – nové funkcie, opravy chýb, známe problémy. Tester overuje, či všetky položky uvedené v Release Notes boli reálne nasadené, či fungujú podľa očakávania a či opravené chyby sa už nevyskytujú.

**3-DS overenie**

(3-D Secure) – bezpečnostný protokol pre online platby kartou, ktorý vyžaduje dodatočné overenie (napr. SMS kód alebo biometria). Tester overuje, či je aktivovaný pre správne typy transakcií, či sa korektne presmeruje na stránku banky a či systém reaguje aj pri zlyhaní overenia.

**SCA**

(Strong Customer Authentication) – silné overenie používateľa podľa európskej smernice PSD2. Zahŕňa dve z troch kategórií: vedomosť (heslo), vlastníctvo (mobil), charakteristika (odtlačok, rozpoznanie tváre). Tester overuje, či sa SCA uplatňuje v správnych scenároch a či je prispôsobené rizikovosti transakcie.

**Webhook PSP**

Automatická spätná notifikácia od platobnej brány (Payment Service Provider) o stave transakcie. Tester overuje, či sa webhook spracuje aj pri oneskorení alebo opakovanom doručení, či podpis webového volania (napr. HMAC) sedí a či sa zmení stav platby v systéme.

**Flapping**

Opakované striedanie stavu služby (napr. *up–down–up*) v krátkom čase. Tester sleduje logy a monitoring, aby sa zistilo, či systém správne rozpoznáva trvalé výpadky, či flapping nespúšťa zbytočné alerty a neovplyvňuje retry logiku.

**Idempotency key**

Jedinečný identifikátor, ktorý zaručí, že opakovaná požiadavka nevykoná tú istú operáciu viackrát (napr. duplicitná platba). Tester overuje, či systém správne spracuje identický request s rovnakým kľúčom ako *safe retry* a neuloží duplikát.

**Transakcia / saga**

Model riadenia viacstupňových operácií – *saga* rozkladá veľkú transakciu na menšie kroky s možnosťou kompenzácie (rollback). Tester overuje, či sa všetky kroky vykonajú atómovo alebo sa v prípade chyby správne vrátia späť.

**TTL koncepty**

(Time To Live) – doba, počas ktorej je údaj alebo záznam platný (napr. cache, token, DNS). Tester overuje, či sa po uplynutí TTL objekt automaticky obnoví alebo vymaže, a sleduje, či sa hodnota TTL neaplikuje nesprávne na iné typy dát.

**Pending payment**

Platba, ktorá bola inicializovaná, ale ešte nemá finálny status. Tester overuje, či sa takýto stav správne zobrazuje používateľovi, či má systém retry mechanizmus a či sa platba automaticky dokončí po prijatí potvrdenia.

**Chargeback**

Proces reklamácie platby iniciovaný klientom, pri ktorom sa suma vracia späť cez kartovú schému. Tester overuje správne označenie transakcie, reakciu systému (blokovanie, refund), logovanie dôvodov a súlad s pravidlami schémy.

**Refund amount**

Suma, ktorá sa má vrátiť klientovi po stornovaní alebo reklamácii. Tester sleduje výpočet refundu, či sa uplatňujú správne poplatky, limity a či refundácia korektne mení zostatok účtu alebo stavu objednávky.

**Race condition**

Situácia, keď výsledok závisí od poradia paralelných operácií. Tester simuluje súbežné požiadavky (napr. dve platby naraz) a overuje, či systém používa správnu synchronizáciu, transakcie alebo zámky.

**ACID**

Vlastnosti databázových transakcií: Atomicity, Consistency, Isolation, Durability. Tester overuje, že transakcie sú buď úplne vykonané, alebo úplne zrušené, dáta zostávajú konzistentné a že nedochádza k strate údajov ani po výpadku.

**Test charter**

Krátky dokument alebo poznámka definujúca cieľ, rozsah a očakávaný výsledok jednej testovacej session (najmä pri exploratívnom testovaní). Tester ju používa na zameranie testov a neskôr na reportovanie výsledkov.

**Defect clustering**

Jav, pri ktorom väčšina chýb vzniká v malom počte modulov alebo funkcií. Tester využíva túto informáciu pri prioritizácii testovania a rizikovej analýze.

**Pesticide paradox**

Situácia, keď tie isté testy už prestávajú odhaľovať nové chyby, pretože boli všetky bežné scenáre otestované. Tester pravidelne aktualizuje testy, aby zachytil nové chyby a pokryl zmeny v systéme.

**Orthogonal array testing**

Metóda návrhu testov, ktorá minimalizuje počet kombinácií vstupov pri zachovaní pokrytia všetkých dôležitých interakcií. Tester ju využíva v prípadoch, kde je veľa vstupných parametrov.

**Error seeding**

Úmyselné vkladanie známych chýb do systému alebo dát, aby sa otestovala efektivita testov alebo tímu. Tester potom vyhodnocuje, koľko seeded chýb bolo odhalených oproti reálnym.

**Defect leakage**

Miera, koľko chýb prešlo z testovacieho prostredia do produkcie. Tester sleduje túto metriku na zlepšenie kvality testovania a určenie slabých miest v procese.

**Test case traceability**

Sledovanie väzby medzi testovacím prípadom, požiadavkou a chybou. Tester overuje, že každá požiadavka má priradené testy a že zistené chyby sú spätne priraditeľné k príslušným požiadavkám.

**Requirement volatility**

Miera zmien požiadaviek počas vývoja. Tester sleduje vplyv týchto zmien na testy, odhady a pokrytie – časté zmeny zvyšujú riziko nedotestovaných častí.

**Test data management**

Proces vytvárania, aktualizácie a čistenia testovacích dát. Tester sleduje, aby boli dáta realistické, bezpečné (anonymizované) a aby sa dali jednoducho obnoviť.

**Self-healing tests**

Automatizované testy, ktoré sa dokážu samy opraviť pri zmene lokátorov alebo prvkov v UI. Tester kontroluje, či sú zmeny korektné a či test neprešiel len „náhodou“.

**Chaos monkey**

Nástroj, ktorý náhodne vypína služby v produkčnom alebo testovacom prostredí, aby sa overila odolnosť systému. Tester sleduje, či sa služby po výpadku automaticky zotavia.

**Synthetic data generation**

Tvorba umele generovaných dát, ktoré simulujú reálne scenáre bez ohrozenia citlivých údajov. Tester overuje ich štruktúru, variabilitu a pokrytie okrajových prípadov.

**Test observability**

Schopnosť testov a systému poskytovať dostatočné informácie pre analýzu výsledkov – logy, metriky, trace ID. Tester sleduje, či testy ponúkajú dostatok dôkazov pre debugging.

**Shadow deployment**

Nasadenie novej verzie aplikácie paralelne s produkčnou, ale bez vplyvu na reálnych používateľov. Tester porovnáva správanie oboch verzií na rovnakých dátach.

**Fault tolerance**

Schopnosť systému pokračovať v chode aj pri výskyte chyby. Tester overuje, či sa procesy obnovia, či sa logujú chyby a či nedochádza k strate dát.

**Recovery testing**

Testovanie schopnosti systému obnoviť sa po výpadku – napríklad po páde servera, strate spojenia alebo chybnom deployi. Tester sleduje, či sa systém reštartuje bez straty transakcií.

**Test environment drift**

Situácia, keď sa testovacie prostredie odlišuje od produkčného (napr. iné konfigurácie, verzie). Tester overuje, či sa prostredia pravidelne synchronizujú a či testy bežia na aktuálnych dátach.

**Data pipeline testing**

Testovanie tokov dát od zdroja po cieľ (ETL, streamy). Tester kontroluje, či sa dáta správne načítajú, transformujú a ukladajú bez straty alebo duplikácie.

**Golden dataset**

Referenčná množina dát, na ktorej sa porovnáva správnosť výstupov systému. Tester ju používa pri validácii dátových transformácií alebo modelov AI.

**Test maturity model**

Rámec na hodnotenie úrovne vyspelosti testovacích procesov v organizácii (napr. TMMi). Tester alebo QA lead ho používa na plánovanie zlepšení – od chaotického testovania po optimalizované a automatizované procesy.

**Exploratory charter**

Dokument, ktorý definuje cieľ, rozsah a predpoklady jednej exploratívnej relácie. Tester v ňom popíše, čo chce preskúmať, aké riziká sleduje a ako bude zaznamenávať výsledky.

**Defect taxonomy**

Klasifikácia chýb podľa ich typu, príčiny alebo dopadu (napr. UI, logika, integrácia, výkonnosť). Tester ju využíva pri reportovaní, aby bolo možné analyzovať trendy a slabé miesta.

**Test strategy**

Vysokoúrovňový dokument popisujúci prístup k testovaniu – cieľ, typy testov, prostredia, nástroje a zodpovednosti. Tester sa podľa nej riadi pri tvorbe testovacieho plánu.

**Test plan**

Dokument, ktorý špecifikuje konkrétny rozsah, termíny, testovacie prípady a zdroje pre dané vydanie alebo projekt. Tester ho vytvára pred spustením testovania a pravidelne aktualizuje.

**Smoke suite**

Zoznam základných testov, ktoré sa spúšťajú po nasadení novej verzie. Tester overuje, že aplikácia „nehorí“ a že základné funkcie sú dostupné ešte pred spustením celej regresie.

**Regression debt**

Nahromadené chýbajúce alebo neudržiavané regresné testy. Tester sleduje, či sú testy aktuálne a či pokrývajú nové funkcionality, aby sa predišlo falošnému pocitu kvality.

**Test flakiness score**

Metrika, ktorá vyjadruje nestabilitu automatizovaných testov. Tester sleduje percento zlyhaní, ktoré sa neopakujú, a hľadá príčiny ako časovanie, sieťové chyby či závislosti od dát.

**Feature toggle testing**

Overovanie správneho fungovania funkcií, ktoré sa dajú zapínať a vypínať cez konfiguračné flagy. Tester overuje, či sú zmeny okamžité a nemajú vedľajšie efekty.

**Contract schema validation**

Testovanie formátu a obsahu správ v integračnom rozhraní (napr. JSON Schema, XML XSD). Tester overuje, že schéma sa nemení nekompatibilným spôsobom a že dáta spĺňajú typy a povinné polia.

**Canary rollback**

Automatické vrátenie nasadenej časti aplikácie, ak nová verzia generuje viac chýb ako predchádzajúca. Tester sleduje správnosť prahov, logovanie a stav po obnovení.

**API throttling**

Mechanizmus, ktorý obmedzuje počet požiadaviek v čase. Tester overuje, či sa správne uplatňujú limity, aké odpovede API vracia po prekročení limitu a či je rate limit spravodlivo rozdelený.

**Security header testing**

Overovanie bezpečnostných hlavičiek v odpovediach HTTP (napr. CSP, HSTS, X-Frame-Options). Tester kontroluje, či sú správne nastavené pre ochranu proti útokom ako XSS alebo Clickjacking.

**Token rotation**

Bezpečnostný mechanizmus, pri ktorom sa autentifikačné tokeny po určitom čase automaticky obnovujú. Tester sleduje, či nový token funguje bez výpadku relácie a či sa starý správne zneplatní.

**Session timeout testing**

Testovanie, či aplikácia správne ukončí reláciu po nečinnosti alebo po uplynutí platnosti tokenu. Tester overuje, že používateľ musí po čase znova prejsť overením.

**Retry queue**

Fronta požiadaviek, ktoré sa majú spracovať opakovane po chybe. Tester sleduje, či sa retry správa podľa politiky (počet pokusov, odstup, dead-letter queue) a či sa nestrácajú správy.

**Data lineage**

Sledovanie pôvodu a transformácií dát v systéme. Tester využíva data lineage na validáciu ETL procesov, audity a kontrolu konzistencie dát medzi zdrojom a cieľom.

**Synthetic transaction**

Simulovaná transakcia používaná v monitoringu – napríklad prihlásenie alebo platba s testovacími údajmi. Tester overuje, že aplikácia reaguje správne a že sa výsledky ukladajú v monitorovacom systéme.

**Non-functional requirement (NFR)**

Požiadavka, ktorá nepopisuje funkciu, ale vlastnosť systému (napr. výkon, dostupnosť, bezpečnosť). Tester overuje, či sú NFR merateľné a validované testami.

**Latency**

Oneskorenie medzi odoslaním požiadavky a prijatím odpovede. Tester sleduje, či latencia neprekračuje definované SLA a či systém správne reaguje pri zvýšenej odozve.

**Error budget**

Množstvo chýb, ktoré si systém môže dovoliť v rámci stanoveného cieľa spoľahlivosti. Tester sleduje využitie error budgetu a spolupracuje s DevOps tímom pri rozhodovaní o release.

**Root cause analysis (RCA)**

Proces zisťovania skutočnej príčiny chyby, nie len jej symptómov. Tester analyzuje logy, vstupy a prostredie, aby navrhol preventívne opatrenia.

**Incident report**

Dokument, ktorý sumarizuje priebeh a dopad incidentu v produkcii. Tester ho používa na spätnú analýzu, identifikáciu slabých miest a návrhy na zlepšenie procesov.

**Test artifact**

Každý dokument, skript alebo výstup, ktorý vznikne počas testovania – od plánu až po report. Tester ich archivuje pre audit, spätné dohľadanie a preukázanie kvality.

**Release candidate**

Verzia aplikácie, ktorá je považovaná za pripravenú na produkčné nasadenie, ak prejde posledným kolom testovania. Tester sleduje stabilitu, známe chyby a rizikové oblasti pred schválením.

**Dead-letter queue**

Špeciálna fronta, do ktorej sa presúvajú správy, ktoré sa nepodarilo doručiť alebo spracovať po viacerých pokusoch.Tester overuje, že systém správne identifikuje zlyhané správy, že sa do tejto fronty dostanú len neúspešné prípady a že obsahuje všetky potrebné údaje na analýzu chyby.  
V rámci testovania sa sleduje aj proces **retry mechanizmu** – či sa správa po opätovnom pokuse spracuje správne alebo zostane v „dead-letter“ stave.Tester kontroluje, či sú správy z tejto fronty auditované, logované a či sa dajú bezpečne znova odoslať po oprave problému.

**Messaging systém**

Systém, ktorý sprostredkúva komunikáciu medzi aplikáciami pomocou výmeny správ (napr. RabbitMQ, Kafka, Azure Service Bus). Tester overuje spoľahlivosť doručovania, poradí, konzistenciu a spracovanie v prípade výpadku. Sleduje aj reakcie na duplicity, stratu správ a správanie pri veľkej záťaži.

**Message acknowledgment**

Mechanizmus potvrdenia prijatia správy prijímateľom. Tester kontroluje, či sa správy označujú ako spracované až po úspešnom spracovaní a či sa pri chybe vracajú do fronty alebo dead-letter queue. Overuje sa aj reakcia systému pri oneskorenom alebo chýbajúcom potvrdení.

**Retry policy**

Politika opakovaných pokusov o spracovanie správy po zlyhaní. Tester sleduje počet povolených pokusov, intervaly medzi nimi, exponenciálne oneskorenie a reakciu po prekročení limitu. Overuje, že správy sa po neúspechu presunú do správnej fronty a nespôsobujú nekonečné cyklenie.

**Message ordering**

Zachovanie správneho poradia správ pri ich spracovaní. Tester overuje, či systém garantuje FIFO (first-in, first-out) spracovanie, či sa poradie nezmení po paralelnom spracovaní a či aplikácia správne reaguje na out-of-order doručenie.

**Poison message**

Správa, ktorá opakovane zlyháva pri spracovaní (napr. kvôli nevalidným dátam alebo chybe v logike). Tester sleduje, či je takáto správa po viacerých pokusoch identifikovaná a presunutá do dead-letter queue, či sa správne loguje dôvod zlyhania a či nedochádza k blokovaniu celej fronty.

**Data drift**

Zmena v dátach, ktoré prichádzajú do modelu, oproti dátam, na ktorých bol model trénovaný.  
Tester sleduje metriky ako rozdelenie hodnôt, korelácie alebo priemerné odchýlky a upozorňuje, keď model začne generovať nepresné predikcie.

**Bias detection**

Detekcia zaujatosti (bias) v dátach alebo modeloch, ktorá spôsobuje nespravodlivé rozhodnutia.  
Tester overuje, či sú trénovacie dáta vyvážené, či model nepreferuje konkrétnu skupinu a používa metriky ako *disparate impact* alebo *equal opportunity*.

**Model explainability**

Schopnosť vysvetliť, prečo model urobil konkrétne rozhodnutie. Tester sleduje výstupy nástrojov ako LIME, SHAP alebo Feature Importance, aby overil, že model sa rozhoduje na základe logických a etických atribútov, nie náhodných korelácií.

**Feature store**

Centrálne úložisko spracovaných atribútov (features), ktoré sa používajú na trénovanie a inferenciu modelov. Tester kontroluje konzistenciu medzi trénovacími a produkčnými dátami, verziu feature setov a správne načítanie hodnôt pri rôznych dátových zdrojoch.

**Synthetic features**

Umelo vytvorené atribúty, ktoré vznikajú kombináciou iných premenných (napr. pomer, rozdiel, log transformácia). Tester overuje, že výpočty sú správne, že feature má prediktívnu hodnotu a že sa generuje konzistentne vo všetkých prostrediach.

**Golden record**

Najpresnejšia a najúplnejšia verzia záznamu po zlúčení dát z viacerých zdrojov. Tester kontroluje deduplikáciu, správne mapovanie polí a konflikt pri rôznych hodnotách. Sleduje, či sa golden record aktualizuje pri zmene vstupných údajov.

**Data skew**

Nerovnomerné rozdelenie dát, ktoré môže ovplyvniť presnosť modelu. Tester sleduje rozptyl, median a distribúciu hodnôt, aby identifikoval nevyváženosť (napr. extrémne časté alebo zriedkavé kategórie).

**Reproducibility testing**

Overovanie, že rovnaký model, kód a dáta dajú pri opakovanom spustení rovnaké výsledky. Tester sleduje verzie knižníc, seed pre generátory náhodných čísel a kontroluje, či pipeline produkuje identické výstupy.

**Pipeline orchestration**

Riadenie a koordinácia krokov v dátovej alebo ML pipeline (napr. pomocou Airflow, Kubeflow). Tester overuje správne poradie úloh, spúšťanie závislostí, opätovné vykonanie po chybe a konzistentné logovanie.

**Monitoring metrics**

Sledovanie metrík modelu po nasadení (napr. accuracy, F1-score, recall, latency, throughput).  
Tester sleduje trend týchto metrík, upozorňuje na pokles výkonu a spolupracuje s dátovým tímom na identifikácii príčin (data drift, zmena správania používateľov).

**Model retraining**

Proces opätovného natrénovania modelu na nových alebo rozšírených dátach, aby sa zachovala jeho presnosť. Tester sleduje, či sa retréning spúšťa pri správnych podmienkach (napr. pri data drifte), či sú dáta validované pred použitím a či sa po retréningu výkon modelu reálne zlepšil.

**Concept drift**

Zmena vzťahu medzi vstupnými dátami a cieľovou premennou v čase. Tester overuje, či systém rozpozná zmenu vzoru správania (napr. iné preferencie používateľov) a či model reaguje správne – napríklad retréningom alebo úpravou váh.

**Feature leakage**

Situácia, keď model počas tréningu používa informácie, ktoré by v reálnom predikčnom čase nemal poznať (napr. budúce údaje). Tester overuje, či dátové pipeline neobsahujú úniky informácií a či nevedú k umelo vysokej presnosti modelu počas testovania.

**Baseline model**

Referenčný model, ktorý slúži ako počiatočné porovnanie pre ďalšie experimenty. Tester sleduje, či nový model dosahuje lepšie metriky ako baseline, a či zmena v algoritme alebo dátach prináša skutočné zlepšenie, nie len štatistickú odchýlku.

**Fairness testing**

Testovanie spravodlivosti modelu – zisťovanie, či model nerozhoduje zaujatým spôsobom voči skupinám používateľov (napr. podľa pohlavia, veku alebo etnika). Tester overuje metriky ako disparate impact, equal opportunity či demographic parity a kontroluje, že model spĺňa etické a právne požiadavky.

**Shadow model**

Paralelný model bežiaci vedľa produkčného, ktorý spracúva tie isté dáta, ale jeho výsledky sa zatiaľ nepoužívajú. Tester porovnáva výstupy oboch modelov, sleduje rozdiely v predikciách a pomáha rozhodnúť, či je nový model pripravený na nasadenie.

**Model versioning**

Sledovanie verzií modelov, ich parametrov, dát a výsledkov. Tester overuje, či sú všetky verzie správne označené, či sa dajú spätne zreprodukovať a či produkčné prostredie vždy používa schválenú verziu modelu.

**Inference latency**

Čas, ktorý model potrebuje na spracovanie vstupu a vygenerovanie predikcie. Tester sleduje oneskorenie pri rôznych typoch dát, zisťuje, či sa latencia zvyšuje pri väčšej záťaži, a porovnáva výsledky s definovanými SLA.

**Model registry**

Úložisko, ktoré centralizovane spravuje verzie modelov, ich metadáta a stav (napr. tréning, schválený, produkčný). Tester overuje, či sa nasadzuje správna verzia modelu, či registry obsahuje auditovateľné údaje (autor, dátum, metriky) a či je integrácia s pipeline bezchybná.

**Disparate impact**

Miera, ktorá ukazuje, či rozhodovanie modelu znevýhodňuje určitú skupinu ľudí (napr. ženy, seniorov). Tester ju používa na vyhodnotenie spravodlivosti modelu – ak má jedna skupina podstatne nižšiu mieru schválenia, model môže byť považovaný za diskriminačný.

**Equal opportunity**

Metrika spravodlivosti, ktorá porovnáva pravdepodobnosť správnych rozhodnutí (true positive rate) medzi skupinami. Tester sleduje, či má model rovnakú šancu správne predpovedať pozitívny výsledok bez ohľadu na príslušnosť ku skupine.

**Demographic parity**

Podmienka, pri ktorej model produkuje rovnaké percento pozitívnych predikcií pre všetky skupiny bez ohľadu na ich vlastnosti. Tester overuje, či model spĺňa túto rovnováhu, alebo či existuje nežiaduce zvýhodnenie jednej kategórie používateľov.

**Airflow**

Open-source nástroj na plánovanie, spúšťanie a monitorovanie dátových pipeline. Tester overuje, či úlohy v DAG (Directed Acyclic Graph) prebiehajú v správnom poradí, či sú chyby zachytené v logoch a či systém správne zvláda zlyhania alebo retry pokusy.

**Kubeflow**

Platforma na správu ML workflowov v Kubernetes. Tester sleduje správne nasadzovanie modelov, validáciu pipeline (tréning, testovanie, deploy) a konzistenciu dát medzi jednotlivými komponentmi.

**Model governance**

Rámec pre správu životného cyklu modelov – zahŕňa ich schvaľovanie, auditovanie, sledovanie výkonu a súlad s predpismi. Tester overuje, či sú modely riadne dokumentované, validované a či existujú pravidlá pre ich údržbu, retréning a vyradenie.

**Drift detection**

Mechanizmus, ktorý automaticky odhaľuje zmenu v dátach (data drift alebo concept drift). Tester sleduje, či systém správne rozpoznáva anomálie, spúšťa upozornenia a zachytáva trendy, ktoré by mohli znížiť presnosť modelu.

**Hyperparameter tuning**

Proces optimalizácie parametrov modelu (napr. learning rate, počet neurónov) s cieľom dosiahnuť najlepší výkon. Tester sleduje, či sú experimenty reprodukovateľné, či sa logujú výsledky a či zmena parametrov prináša stabilné zlepšenie.

**Cross-validation**

Metóda validácie modelu, ktorá rozdeľuje dáta na viac častí (folds) a opakovane trénuje a testuje model na rôznych kombináciách. Tester overuje, či implementácia cross-validácie neobsahuje únik dát, či sa výsledky medzi foldmi výrazne neodlišujú a či metriky odrážajú reálnu generalizáciu modelu.

**Feature selection**

Proces výberu najdôležitejších vstupných premenných pre model. Tester sleduje, či metóda výberu (napr. filter, wrapper, embedded) zlepšuje výkon a či neodstraňuje kľúčové atribúty potrebné pre správnu interpretáciu.

**Model deployment strategies**

Spôsoby nasadzovania modelov do produkcie – napríklad blue-green, canary alebo shadow deployment. Tester overuje, či sa model nasadzuje bez výpadku, či sú metriky sledované po release a či sa dá rýchlo vrátiť k predchádzajúcej verzii.

**A/B model evaluation**

Porovnanie dvoch modelov (A a B) na rovnakých dátach alebo používateľoch. Tester sleduje, či sú skupiny rozdelené spravodlivo, či sa metriky zbierajú správne a či rozdiel vo výkone je štatisticky významný.

**Data validation pipeline**

Automatizovaný proces, ktorý kontroluje kvalitu dát pred ich spracovaním – validuje typy, formát, rozsahy a chýbajúce hodnoty. Tester overuje, že pipeline zastaví spracovanie pri kritických chybách a že loguje nevalidné záznamy pre audit.

**Monitoring dashboards**

Vizualizácie, ktoré zobrazujú stav systému, výkon modelov, latenciu alebo počet chýb. Tester sleduje, či dashboardy zobrazujú aktuálne dáta, či majú nastavené alarmy a či sú metriky zrozumiteľné pre technický aj biznis tím.

**Lineage tracking**

Sledovanie pôvodu dát od zdroja po konečný výstup. Tester overuje, že každá transformácia, zdroj aj verzia datasetu sú zaznamenané, aby bolo možné spätne zreprodukovať výsledky alebo auditovať chybu.

**Concept drift**

Zmena vzťahu medzi vstupnými dátami a cieľovou premennou v čase. Model, ktorý sa naučil vzory na historických dátach, začína poskytovať menej presné predikcie, pretože realita sa zmenila (napr. zmena správania zákazníkov, sezónnosť, nové trendy). Tester overuje, či systém obsahuje mechanizmy na detekciu driftu, či sa pri jeho výskyte spúšťa retréning modelu a či sa výsledky po aktualizácii zlepšujú. Sleduje tiež, či sa drift neskrýva v niektorých atribútoch (features), ktoré sa časom stávajú neaktuálnymi alebo nereprezentatívnymi.

**DMS (Document Management System)**

Systém na správu elektronických dokumentov počas ich celého životného cyklu – od vytvorenia cez schvaľovanie, archiváciu až po vyradenie. Tester overuje, či sa dokumenty správne ukladajú, indexujú, vyhľadávajú a či sa zachováva integrita a verzovanie súborov.

**Metadata**

Dáta o dokumente – napríklad názov, autor, dátum vytvorenia, verzia, oddelenie alebo typ zmluvy. Tester sleduje, či sa metadáta správne generujú, ukladajú a aktualizujú, či sú zoraditeľné a filtrované podľa definovaných polí.

**Workflow**

Postupnosť krokov, ktoré dokument prechádza – napríklad „vytvorenie → kontrola → schválenie → archivácia“. Tester overuje, či systém správne riadi tok dokumentov medzi používateľmi, či reaguje na zmeny stavov a či má ochranu proti zacykleniu alebo vynechaniu kroku.

**Versioning**

Mechanizmus sledovania zmien dokumentu a uchovávania jeho predchádzajúcich verzií. Tester overuje, že pri uložení novej verzie zostávajú staré prístupné, že sa dá zobraziť história zmien a že systém správne označuje aktuálnu verziu.

**Check-in / Check-out**

Funkcia, ktorá zabraňuje, aby viac používateľov upravovalo ten istý dokument súčasne. Tester sleduje, či sa dokument pri check-out zamkne, či sa po uložení správne odomkne a či sa konfliktné zmeny správne detegujú.

**OCR (Optical Character Recognition)**

Technológia na prevod naskenovaného textu alebo obrázka na editovateľný text. Tester overuje presnosť rozpoznávania, podporu jazykov, spracovanie diakritiky a formátovanie textu (riadky, odseky).

**Indexácia**

Proces vytvárania indexov pre rýchle vyhľadávanie v dokumentoch. Tester overuje, či sa nové dokumenty po nahratí dostanú do indexu, či sa výsledky vyhľadávania zobrazujú správne a či systém reaguje aj na synonymá alebo fulltext.

**Digitálny podpis**

Elektronický podpis, ktorý zaručuje identitu podpisujúcej osoby a nemennosť obsahu. Tester sleduje platnosť podpisu, správne overenie certifikátu a reakciu systému pri expirácii alebo zrušení podpisu.

**Časová pečiatka**

Dôkaz o tom, že dokument existoval v konkrétnom čase. Tester overuje, či sa pečiatka správne aplikuje, uchováva a či ju systém dokáže overiť aj po rokoch.

**Archivačná lehota**

Doba, počas ktorej je dokument povinne uchovávaný podľa zákona alebo interných pravidiel. Tester sleduje, či systém správne počíta dátum vyradenia, či dokument po lehote prejde do stavu „archivovaný“ a či sa po schválení môže zmazať.

**Retenčná politika**

Pravidlá, ktoré určujú, ako dlho sa dokumenty uchovávajú a čo sa s nimi po lehote stane (napr. automatické zmazanie, archivácia, anonymizácia). Tester overuje, či sa politika uplatňuje pre všetky typy dokumentov a či ju možno auditovať.

**Audit trail**

Záznam všetkých akcií nad dokumentom – kto ho otvoril, zmenil, schválil alebo zmazal. Tester overuje úplnosť a nemennosť logov, kontroluje formátovanie záznamov a validuje, že audit trail sa nedá manuálne meniť.

**Prístupové práva**

Systém oprávnení určujúci, kto môže dokument čítať, upravovať alebo schvaľovať. Tester testuje kombinácie rolí (napr. používateľ, vedúci, auditor), dedenie práv a obmedzenie prístupu k citlivým zložkám.

**Štítkovanie (tagging)**

Priraďovanie kľúčových slov dokumentom na jednoduchšie triedenie a filtrovanie. Tester sleduje, či sa tagy ukladajú, vyhľadávajú a či systém rozlišuje veľkosť písmen alebo diakritiku.

**Lifecycle dokumentu**

Životný cyklus dokumentu – od vytvorenia cez schvaľovanie a publikovanie až po archiváciu alebo vyradenie. Tester overuje, či každá fáza má správne prechodové stavy a či sa pri zmene fázy nemenia neoprávnene atribúty dokumentu.

**Automatizovaný obeh dokumentov**

Nastavenie pravidiel, podľa ktorých systém automaticky presúva dokumenty podľa stavu alebo obsahu (napr. „faktúry nad 5000 € idú na schválenie CFO“). Tester overuje logiku pravidiel, prioritizáciu a správanie pri výnimkách.

**PDF/A formát**

Štandard pre dlhodobú archiváciu elektronických dokumentov. Tester kontroluje, či systém generuje PDF v správnej verzii (napr. PDF/A-1b), či sú vložené fonty a či dokument zostane čitateľný aj po rokoch.

**Migrácia dokumentov**

Presun dokumentov z jedného systému do iného (napr. pri zmene DMS). Tester overuje úplnosť, integritu, zachovanie metadát a priradenie k správnym workflowom a používateľom.

**Backup a obnova dokumentov**

Proces zálohovania a obnovy súborov a databáz. Tester overuje, že sa zálohy vykonávajú podľa plánu, že obsahujú všetky dokumenty a že sa dajú obnoviť do pôvodného stavu bez poškodenia.

**IaC**

**Terraform**

**APM nástroj**

**Grafana**

**SAST**

**DAST**

**T-shaped model**

**IMS platforma**

**VoLTE**

**OSIX**