Ako testovať výpadok externej služby (napr. platobnej brány)

*Príklad:*  
Predstav si e-shop, ktorý na spracovanie platieb využíva externú platobnú bránu. Zákazník prejde celým procesom nákupu, vloží tovar do košíka, vyplní údaje, zvolí platbu kartou – a v tom sa platobná brána stane nedostupnou. Dôvod môže byť plánovaná údržba, neočakávaný výpadok, sieťový problém alebo chyba na strane poskytovateľa. Ako sa má aplikácia zachovať? Má transakciu zopakovať? Ponúknuť iný spôsob platby? Uložiť objednávku a spracovať ju neskôr?

Takáto situácia je kritická – ak aplikácia reaguje nesprávne, môže to viesť k strate predaja, zmätku u zákazníka alebo dokonca k finančným nezrovnalostiam (napr. peniaze stiahnuté z účtu, ale objednávka neexistuje).

**Ako otestovať:**

1. **Simulácia nedostupnosti API** – vypni pripojenie k testovacej verzii externej služby alebo použi nástroje ako Postman + Mock Server, Fiddler, Charles Proxy či firewall pravidlá, aby si simuloval odmietnutie spojenia, timeout alebo HTTP chyby (napr. 503 Service Unavailable).
2. **Overenie chybových správ** – správa musí byť zrozumiteľná, v správnom jazyku, bez technického žargónu, a musí používateľa informovať, čo môže urobiť ďalej (napr. skúsiť znova alebo zvoliť inú platbu).
3. **Fallback mechanizmy** – over, či aplikácia ponúkne alternatívu (napr. bankový prevod, dobierku) alebo možnosť uložiť objednávku do stavu „čaká na platbu“ a poslať zákazníkovi inštrukcie.
4. **Retry logika** – pri krátkych výpadkoch môže aplikácia skúsiť opakovať požiadavku po pár sekundách alebo minútach. Otestuj, či sa počet pokusov neprekrýva s limitmi API a či sa neblokuje účet alebo karta zákazníka.
5. **Integrita objednávok** – over, že nevzniknú „polovičné“ objednávky (v systéme označené ako zaplatené, ale bez potvrdenia od brány) alebo duplicity. To zahŕňa kontrolu záznamov v databáze aj emailových potvrdení.
6. **Logovanie incidentu** – v logoch by mal byť presný čas, typ chyby, identifikátor objednávky a odpoveď (alebo chýbajúca odpoveď) od brány. Bez týchto údajov je riešenie incidentu pre vývojárov a support zložité.
7. **Scenáre obnovenia prevádzky** – po tom, čo sa služba opäť stane dostupnou, over, ako systém spracuje transakcie, ktoré zlyhali alebo čakali. Má ich spracovať automaticky? Alebo vyžadujú zásah administrátora?
8. **Test rôznych typov chýb** – nie každý výpadok je rovnaký. Otestuj timeouty, úplné odmietnutie spojenia, chybové HTTP kódy, neplatnú odpoveď (poškodený JSON/XML) aj oneskorené reakcie.
9. **Testovanie pri reálnych limitoch** – simuluj vysokú záťaž alebo veľký počet paralelných platieb, aby si videl, či sa chyba nezosilňuje pri špičkách.

Ak sa na toto testovanie pozrieš systematicky, vieš odhaliť veľa kritických slabín ešte pred tým, ako ich zažije reálny zákazník. Výpadky externých služieb sú nevyhnutné – otázkou nie je *či* nastanú, ale *ako* na ne tvoj systém zareaguje.

**#AkoTestovať #QA #Testovanie #SoftveroveTestovanie #TestingTips #Eshop #PaymentGateway #IncidentManagement #APItesting #ResilienceTesting**