

20. Testování, unit testování a dokumentace zdrojového kódu

Testování je technický výzkum produktu, jeho kvality a testuje část nebo celek. Účel tohoto výzkumu je poskytnout zpětnou vazbu a najít chyby. Jedná se také o simulaci běhu.

Často se testování podcení, což většinou vede k tomu, že odškodnění chyby stojí 100-1000x více v pozdějších fázích než ve vývoji.

Testovací cyklus

Plánování testů -> Analýza a příprava testů -> Vykonání testů.

Každá tato část má podčást – reporting. Nahlašování. Účelem je dát programátorovi vědět, co má špatně a co je potřeba opravit.

Plánování testů – Prostě... plánování testů.

Analýza a příprava testů – Začátek testů, začnou se psát například unit testy a začnou se připravovat programy, které testy připraví.

Vykonání testů – Spouštění unit testů a sledování výsledků. Je uvnitř také vyhodnocení testů

FURPS – Functionality Usability Reliability Performance Supportability

Functionality (F) - Funkčnost: zaměřuje se na funkční požadavky na software, jako jsou vstupní a výstupní parametry, algoritmy a procesy zpracování dat.

Usability (U) - Použitelnost: týká se uživatelského rozhraní a uživatelské zkušenosti, jako je snadnost použití, přívětivost a srozumitelnost.

Reliability (R) - Spolehlivost: popisuje požadavky na stabilitu a spolehlivost software, včetně výkonu, odolnosti proti chybám a zotavení z chyb.

Performance (P) - Výkon: zahrnuje požadavky na výkon software, jako je rychlost zpracování dat, čas odezvy a výkonové limity.

Supportability (S) - Podpora: týká se požadavků na podporu software, jako jsou správa konfigurace, udržitelnost, dokumentace a školení.

Dělení testů

Whitebox – Víme, jak program funguje, víme jeho strukturu.

Blackbox – Neznáme vůbec nic o programu, prakticky fungujeme jako klient. Dá se to brát i jako user experience. Testování

Graybox – Kombinace jak toho, tak toho. Známe jenom něco.

Typy testů

Unit testy

Testy na jednotlivé funkce nebo metody v kódu. Testuje se tak, zda jednotlivé části kódu fungují správně a splňují požadavky. Jsou často psány převážně programátorem během vývoje softwaru, jsou takovou součástí dokumentace. Noví programátoři mohou vidět, co funkce vlastně mají vracen. Příklad může být testování jednotlivých funkcí v programu, třeba funkce na vracení součtu čísel.

Integration testy

Testy na spojení více jednotlivých částí kódu, například modulů, tříd nebo služeb. Testuje se, zda tyto části kódu spolupracují správně a propojují se bez problémů. Řeší například, jestli hráč ve hře může dopadnout na zem a vystřelit ze zbraně – kombinuje zem, hráče a zbraň. Na webové stránce například to, jestli se data z formuláře dostanou do databáze.

System testy

Testy na celkovou funkčnost systému nebo aplikace. Testuje se, zda všechny funkce, služby a procesy systému spolupracují a splňují požadavky. Příkladem může být testování nového operačního systému na zcela novém počítači.

Acceptance testy

Testy, které ověřují, zda je software schopen splnit požadavky uživatele nebo zákazníka. Testuje se, zda software splňuje specifikace, funkční požadavky a očekávání uživatele. Příkladem může být testování pomocí beta verze, kde uživatelé dávají feedback a my zjišťujeme, jestli dosahujeme toho, co si uživatel představoval.

Regression testy

Testy, které ověřují, zda změny provedené v kódu nezpůsobily žádné nové chyby nebo problémy. Tyto testy se provádějí opakovaně v průběhu vývoje softwaru, aby se zajistilo, že software zůstává funkční a spolehlivý. Příkladem může být spuštění různých integračních nebo systémových testů pokaždé, co se vydá nová verze.

Performance testy

Testy, které ověřují výkon a rychlost software v různých podmínkách, například při zátěži nebo souběžném užívání. Příkladem může být testování rychlosti stránky pod nátlakem mnoha uživatelů.

Security testy

Testy, které ověřují, zda je software chráněn před různými druhy útoků, jako jsou hackerské útoky, phishing, nebo vniknutí do systému.

Automatizované testování

Testování je spojené s programováním – Například automatizované testování. Často se využívají frameworky, často Javascript, nebo v Pythonu robot framework.

Automatické testy – Robot skočí na nějakou stránku s přihlášením, zadá údaje a sleduje, jestli se zobrazí okénko. Slouží k zjednodušení testování, aby nemusely testy být spouštěny ručně.

API testing – Máme jenom end pointy, testujeme, jestli dostáváme data zpátky a jestli běží. To je možné pomocí jednoduchého skriptu posílající requesty na náhodná data.

Frontend – Sidepress. Když dáme CTRL F v F12, kontrolujeme, jestli tam jsou nějaké konkrétní elementy. Dá se hledat například i konkrétní element podle ID, podle classy a další.

Dokumentace zdrojového kódu

Dokumentace zdrojového kódu slouží k popsání slovy toho, co kód dělá a kde například co najdeme v případě hustého kódu.

Dokumentovat se ovšem dají i přímo metody pomocí speciální notace, kdy následně při najetí na metodu nebo proměnnou můžeme nalézt její popis, v případě funkce či metody vstupní parametry či co metody vrací.

Mluvit asi o svých vlastních zkušenostech s dokumentováním, uživatelským manuálem, jak je důležité mít dobrou dokumentaci atd.

Testování her a verze

Alfa testování – testují lidé z firmy

Beta testování – testují vybraní hráči mimo firmu (Open Beta)