Testování softwaru a QA

Testování softwaru je proces vyhodnocování softwarového systému s cílem zjistit, zda splňuje zadané požadavky a identifikovat případné defekty (bugy). **QA (Quality Assurance)** je širší pojem, který zahrnuje celou strategii předcházení chybám v průběhu celého vývojového cyklu.

1. Úrovně testování (Testing Levels)

V softwarovém inženýrství se testy rozdělují podle toho, jak velkou část systému ověřují:

- **Unit testy (Jednotkové):** Testování nejmenších částí kódu (jednotlivých funkcí nebo metod). Provádějí je sami vývojáři. Jsou velmi rychlé a levné.
- **Integrační testy:** Ověřují, zda různé moduly nebo služby správně spolupracují (např. zda aplikace správně komunikuje s databází).
- **Systémové testy:** Testování kompletního, integrovaného softwaru jako celku. Ověřuje se funkčnost i výkon.
- **Akceptační testy (UAT):** Poslední fáze, kdy software testuje koncový uživatel nebo zákazník, aby potvrdil, že systém splňuje jeho potřeby.

2. Typy testování podle přístupu

Black Box vs. White Box

- **Black Box (Černá skříňka):** Tester nezná vnitřní strukturu kódu. Testuje se pouze na základě vstupů a výstupů (z pohledu uživatele).
- **White Box (Bílá skříňka):** Tester má přístup ke zdrojovému kódu a testuje vnitřní logiku, cesty a podmínky v algoritmech.

Statické vs. Dynamické

- **Statické:** Analýza kódu bez jeho spuštění (code review, kontrola dokumentace).
- **Dynamické:** Testování probíhající za běhu programu.

—

3. Funkcionální vs. Nefunkcionální testy

Typ	Co se testuje	Příklad
Funkcionální	„Co“ systém dělá.	Můžu se přihlásit? Funguje tlačítko „Koupit“?
Nefunkcionální	„Jak“ systém funguje.	Jak rychle se stránka načte? Kolik uživatelů server unese?

Důležité nefunkcionální testy:

- **Zátěžové testy (Load Testing):** Ověření chování systému při vysokém zatížení.
- **Penetrační testy:** Simulace útoků pro odhalení bezpečnostních slabin.
- **Testy přístupnosti:** Kontrola, zda je software použitelný pro lidi se zdravotním postižením.

—

4. Automatizace testování (Test Automation)

U velkých projektů je manuální testování všeho nereálné. Proto se používá automatizace:

- **Regresní testování:** Automatické spouštění testů po každé změně v kódu, aby se zajistilo, že nová funkce nerozbila ty stávající.
- **CI/CD (Continuous Integration/Deployment):** Testy se automaticky spouštějí při každém nahrání kódu do verzovacího systému.
- **Nástroje: Selenium** (web), **JUnit/PyTest** (kód), **Cypress** (E2E testování).

—

5. Reportování chyb (Bug Life Cycle)

Nalezená chyba musí projít určitým procesem:

```
1. **New:** Chyba je nahlášena v systému (např. Jira).
2. **Assigned:** Přidělena vývojáři k opravě.
3. **Fixed:** Vývojář chybu opravil.
4. **Retest:** Tester ověří opravu.
5. **Closed:** Chyba je definitivně vyřešena.
```

[Image of Bug Life Cycle flow chart]

Související články:

- Metodiky vývoje (Scrum, Kanban)
- Programovací jazyky a syntaxe
- Verzování kódu a Git

Tagy: dev qa testing automation bug software_quality ci_cd

From:
http://serviceit.cz/ - **IT ENCYKLOPEDIE**

Permanent link:
http://serviceit.cz/doku.php?id=it:dev:testing

Last update: **2026/01/02 13:42**