..

TDD (Test-Driven Development)

TDD není technika testování, ale technika **návrhu** softwaru. Nutí vývojáře přemýšlet o zadání a rozhraní (jak se bude kód používat) dříve, než začne řešit samotnou implementaci (jak to bude fungovat uvnitř).

Cyklus Red – Green – Refactor

TDD probíhá v neustále se opakujícím krátkém cyklu tří fází, který trvá obvykle jednotky minut:

```
1. **RED (Červená):** Napíšete malý [[unit_testing|jednotkový test]] pro funkci, která ještě neexistuje. Spustíte testy – tento nový test musí selhat (svítit červeně). Tím ověříte, že test skutečně něco testuje a neprochází jen náhodou.
2. **GREEN (Zelená):** Napíšete nejjednodušší možný kód (klidně i "špinavý" nebo s napevno zadanou hodnotou), aby test prošel (svítil zeleně). Cílem je mít funkční kód co nejrychleji.
3. **REFACTOR (Refaktorování):** Nyní, když máte jistotu, že kód funguje (test je zelený), jej vyčistíte. Odstraníte duplicity ([[dry_princip|DRY]]), zlepšíte názvy a strukturu. Testy po každé změně znovu spustíte, aby byla jistota, že jste nic nerozbili.
```

Výhody TDD

- **Vysoké pokrytí testy:** Jelikož kód vzniká jen jako odpověď na test, je téměř 100% kódu pokryto automatizovanými testy.
- **Čistší design:** Programátor je nucen psát kód, který je snadno testovatelný, což vede k menším třídám a dodržování SOLID principů.
- **Méně chyb:** Většina logických chyb je odhalena okamžitě během vývoje, nikoliv až testery nebo zákazníky.
- **Odvaha ke změnám:** Díky síti testů se vývojář nebojí provádět velké změny v kódu (refaktorovat), protože okamžitě vidí dopad své úpravy.

Nevýhody a mýty

- **Pomalejší start:** Psaní testů zabere na začátku čas. Z dlouhodobého hlediska se však čas ušetří díky méně chybám a snazší údržbě.
- **Není pro všechno:** TDD se hůře aplikuje na UI (uživatelské rozhraní), složité integrace nebo průzkumný vývoj, kde programátor ještě neví, jak bude výsledek vypadat.
- **Údržba testů:** Pokud se změní zadání, musí se upravovat nejen kód, ale i testy.

Srovnání: Tradiční vývoj vs. TDD

Fáze	Tradiční vývoj	TDD (Vývoj řízený testy)
1. Krok	Návrh a psaní kódu.	Psaní selhávajícího testu.
2. Krok	Ruční zkoušení / Psaní testů.	Psaní kódu pro splnění testu.
3. Krok	Oprava chyb.	Refaktorování pod dohledem testů.
Důraz	Na rychlost prvního dodání.	Na kvalitu a dlouhodobou udržitelnost.

Související pojmy: Unit Testing, Refaktorování, SOLID principy, DRY princip, Technický dluh.