

# Testovanie softvéru podľa ISTQB

Učebná pomôcka od [ITnetwork.sk](https://www.itnetwork.sk) - Prehľad základnej syntaxe

## Základy testovania

Testovanie je proces overovania **spravnosti** a **kvality** softvéru.

### Dynamické testovanie

Prebieha **za reálneho behu aplikácie** (manuálne či automatizovane) a slúži na overenie funkčnosti aj na odhalenie chýb.

### Statické testovanie

Revízia, kontroly a analýzy kódu či dokumentácie **bez spustenia aplikácie**. Cieľom je čo najskôr odhaliť chyby, napríklad v požiadavkách, návrhu alebo kóde.



### Ciele testovania

1. Odhaliť defekty (chybné funkcie, bezpečnostné slabiny).
2. Predísť chybám (**QA** - preventívny prístup).
3. Posúdiť kvalitu produktu (**QC** - nápravny prístup).
4. Overiť, že systém plní potreby používateľa.

### Quality Assurance (QA) vs. Quality Control (QC)

**Quality Assurance (QA)** sa zameriava na procesy a metodiky, teda na overovanie, že softvér vzniká správnym spôsobom (termín **verifikácia** – "Are we building it right?").

**Quality Control (QC)** sa zameriava priamo na hotový produkt a jeho kvalitu, teda na overovanie, že softvér plní očakávania užívateľov (termín **validácia** – "Are we building the right thing?").

## Princípy testovania

- Testovanie ukazuje prítomnosť defektov, nie ich neprítomnosť
- Kompletné testovanie nie je možné - prioritizácia a vhodné techniky
- Včasné testovanie = úspora času aj peňazí
- Zhlukovanie defektov – väčšina chýb sa nachádza v niekoľkých kľúčových oblastiach
- Testy sa opotrebovávajú – je potrebné ich udržiavať a aktualizovať
- Závislosť na kontexte – prístup k testovaniu sa líši podľa projektu
- Neprítomnosť defektov neznamená, že je produkt použiteľný

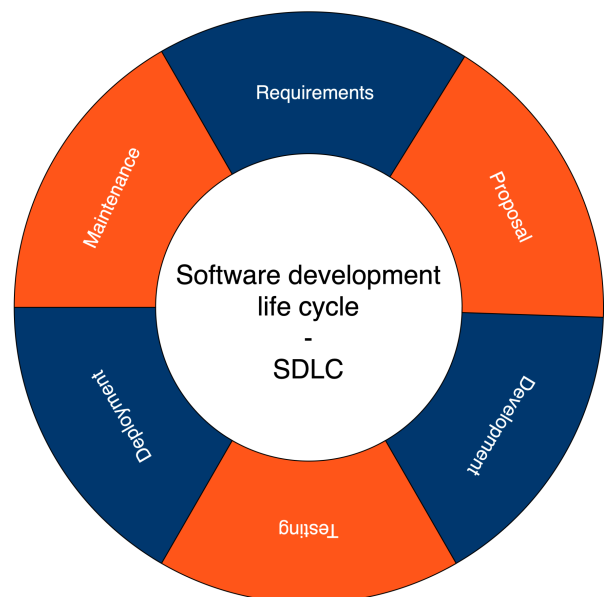
## SDLC (Software Development Life Cycle)

### Modely vývoja

- Sekvenční (napr. **waterfall**) – jasné fázy (analýza, návrh, implementácia, testovanie, nasadenie).
- Iteratívne – cykly vývoja, priebežné zlepšovanie.
- Inkrementálne – pridávanie nových funkčností v malých krokoch.
- Agilné prístupy (**Scrum, Kanban, XP, TDD, BDD, ATDD**) – kladú dôraz na rýchlu spätnú väzbu a neustále zlepšovanie.

### Tester v SDLC

- V sekvenčných modeloch tester kontroľujú dokumentáciu v raných fázach, testovanie prebieha až vo vyhradenej fáze.
- V iteratívnych a agilných modeloch testujú priebežne (časti produktu, inkrementy) a udržiavajú **regression testing** (opakované testovanie, či nové zmeny nepoškodili predtým funkčné časti).



## Testovacie činnosti

- **Test planning**
  - Definícia cieľov testovania, prístupu a harmonogramu.
  - Stanovenie priorít, rizík, prostriedkov.
- **Test monitoring & control**
  - Priebežné sledovanie stavu testov (metriky, reporty).
  - Úpravy plánu testovania podľa potreby.
- **Test analysis**
  - Analýza požiadaviek a špecifikácií.
  - Identifikácia testovacích podmienok a potenciálnych defektov.
- **Test design**
  - Tvorba testovacích scenárov, testovacieho prostredia a dát.
- **Test implementation**
  - Príprava testovacích sád, skriptov a procedúr.
- **Test execution**
  - Spustenie testov, vyhodnotenie výsledkov, reportovanie chýb.
- **Test closure**
  - Archivácia testware, záverečný report, zistenie ponaučenia.



## Testware a report o defekte

**Testware** je **súhrn výstupov z testovacích činností** (napr. test plány, skripty, defekt reporty, protokoly).

**Report o defekte** zaznamenaný počas dynamic testingu obvykle obsahuje **jedinečný identifikátor, názov a zhrnutie anomálie, dátum, organizáciu a autora reportu**. Ďalej zahŕňa **informácie o testovanom objekte a prostredí, kontext defektu, popis zlyhania** umožňujúce jeho reprodukciu a vyriešenie, **očakávané vs. skutočné výsledky, závažnosť a prioritu opravy, stav defektu** a prípadné **odkazy**.

## Úrovne testovania

- **Unit testing** - testovanie jednotlivých komponentov izolovane (obvykle vývojári).
- **Component integration testing** - Test interakcií medzi modulmi/komponentmi.
- **System testing** - test celého systému podľa definovaných požiadaviek (funkčný aj nefunkčný).
- **System integration testing** - Overenie integrácií s externými systémami/službami.
- **End-to-End (E2E) testing** – Overenie celého užívateľského procesu naprieč systémom a jeho integráciami, simulujúce reálne scenáre použitia.
- **Acceptance testing** - Posledná fáza, často zapojení koncoví používatelia (UAT, beta test)

## Typy testov

- **Functional testing** - Kontrola funkčných požiadaviek.
- **Non-functional testing** - Výkon, záťaž, bezpečnosť, kompatibilita, udržovateľnosť, použiteľnosť (ISO/IEC 25010)
- **Black-box testing** - Zamerané na vstupy a výstupy, **bez znalosti vnútornej logiky kódu**, napr.: Equivalence Partitioning (EP), Boundary Value Analysis (BVA), Decision Table, State Transition.
- **White-box testing** - Testovanie **s detailnou znalosťou kódu**, meria pokrytie príkazov a vetiev (Statement & Branch Coverage), zameriava sa na vnútornú štruktúru a logiku programu.
- **Experience-based testing** - Exploratory testing, Checklist-based testing, Bug Estimation.



**itnetwork.sk**

Učíme národ IT