

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Metodika testovania

Tímový projekt – Stratos FIIT

2016

Jakub Findura

1 Úvod

Táto metodika je určená členom tímu, ktorí v rámci svojich úloh tvoria zdrojový kód niektorej zo súčastí systému alebo sú zodpovední za vykonávanie testov. V rámci tímu existuje niekoľko úrovní testovania: jednotkové testy, systémové testy, akceptačné testy. Väčšina testov súvisiaca s hardvérovými súčast'ami systému sú nefunkcionálne testy.

1.1 Roly

Tester – Člen tímu, ktorý je zodpovedný za vykonávanie testov. Je zodpovedný za vykonanie testu, jeho zdokumentovanie a v prípade chyby jej nahlásenie. Postup je v metodike chýb.

Vývojár – člen tímu, ktorý vytvára zdrojový kód. Je zodpovedný za tvorbu jednotkových testov k danému kódu.

1.2 Súvisiace metodiky

- Metodika riadenia
- Metodika chýb

2 Procesy

2.1 Stratosferický modul

Testovaniu servisného modulu stratosférického balóna je venovaná väčšia pozornosť, keďže prípadnú chybu (softvérovú alebo hardvérovú) nie je možné počas letu opraviť a môže dôjsť k neúspešnému ukončeniu letu.

2.1.1 Funkcionálne testovanie

Najdôležitejšie je testovanie zdrojového kódu, ktorý nie je závislý od komunikácie s hardvérom. Testovanie je formou jednotkových testov. Za tvorbu a vykonávanie týchto testov je zodpovedný člen tímu, ktorý implementuje danú funkcionálnosť.

2.1.2 Nefunkcionálne testovanie

V rámci nefunkcionálneho testovania je nutné testovať výnimočné prípady, ktoré môžu nastať pri lete, hlavne v súvislosti s hardvérom. Nefunkcionálne testy sú vykonávané na hotovom prototypu. Dôležité oblasti testovania sú komunikácia modulu s pozemným strediskom, spoľahlivosť modulu v extrémnych podmienkach (nárazy, teplota), výdrž batérií.

2.2 Serverová aplikácia

Testovanie servera je formou jednotkových testov. Za tvorbu testov je zodpovedný člen tímu, ktorý implementuje danú funkcionálnu priradenú v úlohe. Pred uzavretím úlohy je potrebné aby všetky testy prebehli úspešne.

Súčasťou nasadenia novej produkčnej verzie je vykonanie všetkých funkčných testov členom tímu zodpovedným za nasadenie a členom tímu zodpovedným za prehliadku tímu.

2.2.1 Vykonávanie jednotkových testov

Framework Flask, ktorý je použitý v serverovej aplikácii, poskytuje moduly pre tvorbu testov a ich vykonávanie.

```
# balon/balon_test.py
import unittest
from balon import main, db

class BalonTestCase(unittest.TestCase):

    def setUp():
        ...

    def tearDown():
        ...

    def test_metoda_sub(self):
        a = 3
        b = 3
```

Ukážka 1 Vzorový súbor pre jednotkové testy

Test spustíme príkazom **python balon_test.py**

```
C:\dev\TP\server>python balon_test.py
2016-12-13 01:01:48,105 - Balon Logger [DEBUG] __init__.<module>(): Database
Path: C:\dev\TP\server\balooooooooon.sqlite3
2016-12-13 01:01:49,589 - Balon Logger [DEBUG] __init__.<module>(): Starting
flask app __init__.py
main.<module>(): Starting flask app main.py

....

-----
Ran 5 tests in 0.143s

OK
```

Ukážka 2 Výpis po úspešnom vykonaní všetkých testov

2.2.2 Systémové a akceptačné testy

Pred odovzdaním prototypu je vykonané systémové a akceptačné testovanie. Systémové testovanie vykonávajú zodpovední členovia tímu. Cieľom je overiť funkčnosť a správny beh webovej aplikácie. Akceptačné testy sú vykonávané spolu so zákazníkom. Cieľom je overiť, či implementovaný systém spĺňa všetky zadané požiadavky.

2.3 Dokumentovanie testovania

Úspešné výsledky funkcionálnych testov nie je potrebné dokumentovať. Systémové testy sú vykonávané spolu s prehliadkou kódu, ktorého schválenie znamená aj úspešné vykonanie testov. Nefunkcionálne testy sú dokumentované v rámci dokumentácie inžinierskeho diela.

Pokiaľ je objavená chyba alebo iný nedostatok, je potrebné ju zdokumentovať. Procesy manažmentu chýb opisuje metodika chýb.