



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

DevOps, Barkochba szoftver változó adatbázisának tesztelése

Peterle István Soma III. évf, (BSc) mérnökinformatikus szakos hallgató
Konzulensek: Micskei Zoltán, Rath Istvan, Szatmáti Zoltán
Témalaboratóriumi összefoglaló
2018/19 I. félév

A félév elején a heti előadások keretein belül megismerkedtünk néhány DevOps-os technológiával. Ezeket a technológiákat integráltuk a saját projectekbe és kialakítottunk egy automatizált fejlesztési környezetet.

A félév második felében főleg a szoftver tesztelésben mélyedtem el. Egy általam készített barkochba program tesztelésével foglalkoztam. A szoftver képes kitalálni, hogy a felhasználó mire gondol. Abban az esetben, ha rosszul tippelne, a felhasználó segítségét kéri és bővíti az adatbázisát.

A szoftver az adatait egy objektumokból álló fában tárolja, amik egy kérdést vagy egy választ reprezentálnak. A fának a levelei a válaszok, és a felhasználói bemenet után, a fa ágain a szoftver a futása során mindig eljut egy lehetséges válaszig.

Egységtesztekkel tudtam ellenőrizni a szoftver moduljainak a helyes működését külön-külön. Viszont az egységtesztek, a tesztek lefutása előtt mindig létrehoznak egy fix adatstruktúrát amin egy-egy modult vizsgálnak. Ahhoz, hogy a rendes adatbázist tesztelni tudjam másfajta tesztet kellett írnom.

Ez a tesz, a teszt osztály által létrehozott adatbázis helyett a valódi adatokat használja, azokat melyek a szoftver futása során változhatnak. Ugyanúgy, mint az egységtesztek kapnak egy minta felhasználói bemenetet és egy elvárt választ. Mivel az adatbázis változik a tanulás során, a fa struktúra levelei egyre mélyebbre kerülhetnek, és egy minta bement után lehetséges, hogy nem érünk el egy levélhez, így a teszt hibával tér vissza akkor is ha a keresett válasz benne van a fában. Ennek a problémának a megoldására, azt a módszert találtam, hogy a bement után a teszt egy DFS algoritmussal bájárja a részfat, ami alatta maradt és egy listába kigyűjti a leveleket, amik lehetséges válaszokat jelentik. Ezzel a listával hasonlítja össze az elvárt eredményt, így a teszt akkor is megtalálja a keresett választ, ha az adatbázis változott, de a bement nem.

A félév végére létrehoztam egy automatizált fejlesztési környezetet, amiben a git-es változások után automatikusan végrehajtódnak a buildelési és a tesztelési feladatok, és értesítés küld a teszteredményekről. Ezt egy AWS EC2-es szerverre telepített Jenkins CI eszközzel oldottam meg.