



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Méréstechnika és Információs Rendszerek Tanszék

## Automatikus tesztelés

**Kiss Dániel III. évf, (BSc) mérnökinformatikus szakos hallgató**

**Konzulensek: dr. Micskei Zoltán adjunktus, MIT**

**Honfi Dávid doktorandusz, MIT**

**Rendszertervezés szakirány**

**Témalaboratórium összefoglaló**

**2016/17. I. félév**

A témalaboratórium keretei között az automatikus tesztelés alapjaival ismerkedtünk meg részletesen, négy fős csoportban dolgozva, Németh Zsolttal, Reischl Barnabással és Szántó Tamással.

Az első héten, az ISTBQ (International Software Testing Qualifications Board) részletes összefoglalóját elolvassva ismerkedtünk meg a szoftvertesztelés fontosságával, továbbá a félév során használt programozási nyelvet is ekkor választottuk. Németh Zsolt és én **C#**, míg Reischl Barnabás és Szántó Tamás a **Java** nyelven folytatták tovább a munkát.

Az első feladat, egységteszt készítése volt, melyhez egy nyílt GitHub projektet használtunk, amely matematikai algoritmusokat valósított meg. A tesztelés során néhány algoritmust kiválasztva, azok működésének helyességét vizsgáltam. Az egységtesztelés során, számos hibát felismertem a kódban, többek között egy szöveg betűit permutáló algoritmus helytelen működését, illetve néhány rendező algoritmus is nagy elemszámú tömb rendezésekor hibás működést eredményezett.

A következő feladatban, a Visual Studio-ban megtalálható Microsoft Pex eszköz segítségével generált tesztekkel hasonlítottam össze a korábban általam készített tesztesetekkel. Az eszköz használata során azt tapasztaltam, hogy az eszköz elsősorban a kódlefedettséget próbálja biztosítani, és önállóan nem garantálja az elvárt (helyes) működést, ezen „hibák” kiküszöbölése a tesztelő feladata.

Az egységtesztelés és az automatikus tesztgenerálás megismerése után, a tesztelendő kódban lévő függőségek miatt fellépő problémák (pl.: más objektumtól való függés, adatbázistól való függés, stb.) megoldásának módjaival ismerkedem meg. A probléma megoldására, többek között a Visual Studio-ban található Microsoft Fakes, a Moq és az NSubstitute eszközöket használtam, ezzel betekintést nyerve a tesztelés során használt úgynevezett mockolásba, mock készítésbe. Mock objektumok segítségével a tesztelendő kód függőségeinek viselkedését szimuláltam, vagy annak nehezen elérhető állapotait állítottam elő.

A félév végén a GUI (Graphical User Interface) teszteléssel is foglalkoztam. A felhasználói felület tesztelése során, egy grafikus számológép alkalmazás működését teszteltem, a Visual Studio Coded UI segítségével. A GUI tesztek, során a futó szoftveren végrehajtott interakciókat rögzítettem, és értékeltem az alkalmazás működését.

A témalaboratórium nagy rálátást adott a szoftvertesztelés alapvető lépéseire, és rengeteg, a későbbi tanulmányok során felhasználható ismeretet biztosított, amely nagy segítséget nyújthat az önálló laboratórium és szakdolgozat elkészítéséhez.